概念設計に対する計算機援用知的支援に関する研究(Dissertation_全文)

AUTHOR(S):

川上, 浩司

CITATION:

川上, 浩司. 概念設計に対する計算機援用知的支援に関する研究. 京都大学, 1994, 博士(工学)

ISSUE DATE:

1994-01-24

URL:

https://doi.org/10.11501/3074837

(2)

概念設計に対する計算機援用知的支援に関する研究

川上 浩司

目 次

第 1 章

序論

本研究は, 物理的構造物に対する概念設計の計算機支援環境について考察したものである。

設計過程のような, 極めて多様な内容をもつ対象を取り扱う際に有効な方法は, 問題構造や処理形態に注目して解析し, 特徴的な処理のパターンによって設計過程を分類することである。このような設計モルフォロジー (形態学) の立場からは, 設計活動はつぎのような一連の過程に分類される [1]。

1. **設計要求の把握**　設計過程における第一歩であり, 設計の目標を明確に与える段階である。実際の設計では, さらに目標とする機能を技術的にブレークダウンして, 設計仕様として与えられる。

2. **概念設計**　設計仕様として与えられた設計要求に合う設計案 (設計モデル) を作り出す過程である。

3. **基本設計**　概念設計の過程で得られた結果を具体化する過程であり, 「機能設計」または「具象化設計」とも呼ばれる。製品の基本的形態や構造方式の決定, 主要目, 主要構成部品の要目・仕様・使用材料などの決定, 基本レイアウトなどの作成を行う過程である。

4. **詳細設計**　基本設計の結果をもとに, さらに詳細構造, 詳細形状, 詳細寸法, 詳細レイアウトなどを決定する過程である。

5. **生産設計**　詳細設計の過程までに得られた結果をいかにして具体化すべきか, すなわち生産技術・工程計画などを検討する過程である。詳細設計までの過程は, 生産の内容が検討されても, 中心課題は製品としての機能の実現であったのに対し, 生産設計においては, 生産のための情報作成に重点が置かれる。

1 の過程は, 現代社会の価値観の多様化や製品自体の大形化, 大量生産化, 機能の高度化などのためにますます重要になってきている。この過程に対する計算機支援は, 「なぜ

このような設計をするのか」を論じる意思決定支援システムの範疇であり,「いかにこのような設計をするのか」を論じる設計支援システムでは取り扱わない。

4, 5 の過程は, 設計行為の内容を手続きとして記述することが比較的容易であり, またそれぞれ独立した個別処理である。さらに, 設計対象の記述も幾何学的な寸法の記述で対応できることが多く, 実際にこの段階を支援する計算機システムが多数提案・商品化されている。その多くは単純な描画機能だけでなく, 3 次元イメージ表示, 形状のパラメトリックな変更, 面積や図心などのパラメータ計算, および NC データの自動生成などの, 計算機の数値演算能力を利用した機能を備えている。

これに対して, **2** および **3** の過程は, 非定形で数値処理とはなじまず, 設計者との協調処理が必要な部分であるため, 知識情報処理技術の導入が期待されている。**3** の基本設計過程は, **2** の概念設計過程において選定された設計案に対して, 基本的形態や部品の仕様を決め, 機能の面から具体化する過程である。この過程では, 設計モデルを設定したのちに, 機能や強度などを確認するための工学的解析と評価, および評価結果に基づく設計モデル案の修正が繰り返される。強度解析, 空間的干渉の有無の確認等のような工学的解析は, 部分的に数値処理が可能であり, 自動化されつつある。

以上の考察により, 近年の製品設計の高度化・効率化・省力化に対する要求に応えるためには, 人間の創造的活動である設計の上流過程 **2, 3**, とくに **2. 概念設計**の支援が重要であると考える。

本研究は, 以上の背景のもとで, 物理的設計物の概念設計に対する, 知識工学的手法による計算機支援環境について検討したものであり, 設計知識の**表現**・**獲得**・**利用**方法の提案で構成されている。

設計知識の表現　概念設計に対する知識情報処理を困難にしている理由は, 関連知識の範囲が事実上設定できないこと, 表層的な知識のみでは設計対象を捉えきれず, 物理的・数理的法則などの基本原理に関する深層知識が必要であること, さらに設計活動が高度に創造的な思考プロセスであり陽に表現し難い知識が存在することなど, 設計知識に関連した問題が多い。

CAD の分野では, 従来のアプローチが設計対象物の幾何形状という表層的な側面だけを捉えているに過ぎないという反省から, 幾何情報だけではなく, 様々な構造属性を含むモデルが種々提案されている。しかし, これらの設計対象物モデルは, あくまで対象の構造のみに注目したものに留まっている。

物理的構造物, すなわち物理法則や現象に注目して構成された構造物は, ある機能の達成を目的として, その構造と機能が, その背後にある物理法則や現象で媒介されることによって構成されたものと捉えることができる。したがって, 現在の CAD システムに見

られる, 設計対象の構造属性に注目したモデルでは, 設計対象物の構造に関する処理は可能であるが, 構造を規定する機能, および機能と構造を関連づける原理 (物理法則) に対する情報処理という, 設計の本質的な部分を扱うことができない。

そこで本研究では, **価値工学 (Value Engineering: VE)** の視点を導入して, 機能・物理法則・構造の陽な取り扱いが可能な設計対象モデルを提案する。

従来より, 価値工学の分野では, 製品やシステムにおける機能と構造の関係の明確化が, 設計, 製造の各段階におけるコスト低減を達成するための不可欠の課題として追求されている [2]。その代表的な手段として, 対象とする製品やシステムが, 目標とする機能 (基本機能) の達成の手段としてどの様な下位機能群を用いているか (機能構成), それらの機能は製品あるいはシステムのどの部分構造により実現されているのか (機能と構造の結びつき) を**機能系統図**と呼ばれるダイヤグラムとして整理する, いわゆる**機能分析法 (Functional Analysis)** が用いられている。機能系統図に表現された機能構成や機能と構造の連関は, 構造だけでなく機能にも注目した設計知識を反映したものであり, 概念設計に有益な情報を提供する。

機能系統図は, つぎのような形で設計事例の成り立ちを説明づける。すなわち, 設計事例は, その基本機能 (G) (目標) を達成するための手段 (副目標) が一連の機能 (F) として選択され, これらの機能を達成する構造属性 (S) が選択され, 最後におのおのの構造属性を有する実体 (R) が選択された結果の生成物であると解釈 (説明) される。

一般に, 人間の合目的選択行為は, teleological (目標)・causal (因果)・economical (経済) という三つの合理性に従うと考えられている [3]。設計活動をこのような合理性に従う連続的選択行為であると捉えるとき, G に対する F の選択は目標合理性を反映したプラン型知識 (P), また, F に対する S の選択は因果合理性を反映した物理因果法則 (L), さらに, S を実現する R の選択は経済合理性を反映した実装化知識 (I) によって根拠づけられる。

以上の考察により, 設計物の成り立ちを説明する知識として G, (P), F, (L), S, (I), R の七つの階層から成る**階層型設計知識**を提案する [4]。この設計知識を用いることによって, 物理因果関係 (L) を原理的な知識とした**物理因果連鎖**による設計対象物の把握が可能となる。

さらに, これらの設計過程および設計対象物を計算機上に記述する方法について検討を加える。機能の表現に関しては, 価値工学の分野の表現法である「熱を出す」,「位置を決める」など「(目的語) を (動詞) する」という表現形式を採用する。このような抽象度の高い表現形式にした場合, 機能の意味内容が一義に定まらないために, 機械的な取り扱いが困難となる。そこで, 設計知識に統一的な表現形式を与え, 計算機上での表現および操作を可能にするものとして, **概念依存理論** [5] を導入する。概念依存理論は, 少数の

基本となるプリミティヴな概念の組合せによって言語の背景にある意味を表現し, 意味論的に自然言語文を理解する枠組である。

機能表現動詞を概念依存理論に準じて分析し, 有限個のプリミティヴな概念の組合せで表現することによって, 機能表現間の意味的重複・包含関係を明らかにすることができる [6]。これによって, 統一的な表現形式を持った機能を計算機上に記述すること, 標準化された解析結果を得ること, および推論規則の導入が可能となる。

構造の表現に関しては, 実体は属性によって認識されるとする, 一般設計学における認識公理 [7] に基づいて, 部分構造 (構造素と呼ぶ) とその関係, および個々の構造の属性に注目した意味ネットワーク表現形式を提案する [8]。

設計知識の獲得 近年, 設計支援のための知識情報処理手法が種々提案されているが, それを実装した計算機システム, いわゆる設計エキスパートシステムは実験段階に留まっている。その理由として, 設計知識を書き下して設計知識ベースを構築することの困難性, すなわちエキスパートシステム構築における "知識獲得ボトルネック" が指摘されており, エキスパートと知識エンジニアによる手作業の知識獲得に代わり, 計算機によって支援された半機械的な知識獲得の必要性が認識されている。本研究では, 既存の設計物を解析し, その設計物が内包する機能達成のための種々のレベルの設計知識を抽出する二つの手法を提案する。

知識獲得の第一の手法として, **EBG**(Explanation-Based Generalization) [9] を応用して, 一つの設計事例において, その基本機能が下位の機能群にどのように展開されているか, 各末端機能がどのような物理因果連鎖により達成されているか, 各物理現象がどのような構造・属性に基づいているかの**説明づけ (“how” explanation)** を階層型設計知識を用いてトップダウン的に行い, この説明の履歴から, 機能・物理現象・構造の各々のレベルにおける設計知識を一般的・操作的な形で抽出する方法を提案する [4][10]。

EBG 法は, ある事例 (Training Example) が目標概念 (Goal Concept) の正事例であることを, あらかじめ用意されている知識 (領域知識: Domain Theory) を用いて説明し, その履歴から目標概念に対して操作性規範 (Operationality Criterion) を満たす記述を一般化された形で獲得する手法である。目標概念として既存設計物の基本機能, 事例として既存設計物の表層的な構造記述, 領域知識として機能構成・機能と構造の連関に関する知識を与えると, 説明の履歴は価値工学における機能系統図と見なすことができ, 機能系統図から解析対象に依存しない一般的な設計知識を自動的に獲得する手法を実現することができる。

本研究では, 提案したシステムを, ホーン節論理を基礎とした論理プログラミング言語 prolog のメタインタプリタとして実装した。これは, EBG に対して指摘されている一般の問題点を解消するための機能を備えている。すなわち, 設計者の意図や設計局面に応じて変化する操作性規範に対応した形で, 単一事例から種々の設計知識を獲得する機能をもつ。また, 知識獲得のために実在事例を解析することが本質的でないという問題点に対しては, 決定関係の論理的記述である決定則 [11] の特殊化による知識獲得を, EBG の枠組で自動的に実行する機能をもつ。

決定則は類推を正当化する方法の一つとして提案された知識表現形式である。厳密な含意則の入手は困難であるが, ある種の決定要因と被決定項目の関係が明らかである対象領域において, その決定関係を決定則によって記述すれば, それから事例記述を加えて含意則を生成することが可能である。したがって, 解析に基づく学習手法への決定則の導入は, 設計知識ベースの作成を容易にするとともに, 説明に基づく学習手法に代表される演繹的な知識生成手法における事例情報の冗長性問題を解決する。

知識獲得の第二の手法として, 既存の設計物を解析して設計者の意図を抽出する方法を提案する [12]。一般に, 一つの機能を達成する構造 (設計解) は複数考えることができる。そこで, 多くの選択枝の中からなぜ一つの設計解を設計者が選んだのか (**“why” explanation**) を, 複数機能間の干渉回避という側面から解析し, メタプラン的な設計知識を導出する。

この解析は, 解析対象事例が公理論的設計アプローチ [13] における設計公理を満たす構造をもつことを前提としている。公理論的設計アプローチは, 設計活動において常に従うべき公理が存在するという仮説を立て, 全ての意思決定局面でこれに従おうとするものである。実在事例がこの公理を満たす (複数の機能がお互いに干渉することなく達成されており, かつそのような構造の中で最もシンプルなものである) ことを前提とすると, 一旦単一の機能だけを達成するように構造を最小化することによって他の機能との干渉を顕在化できること, および構造付加により干渉を回避できることが保証される。この際の非単調な機能回復過程の履歴として, 複数機能達成のためのメタプラン的な設計知識を導出する。機能達成の道筋は, **真実性維持機構** (Truth Maintenance System: **TMS**) [14] を用いた対象記述を導入することによって, システマティックに追跡することができる。

設計知識の利用 本研究で提案する手法によって獲得される設計知識を用いて設計活動を支援する手法を検討する。

一般に, システム設計案を生成する方法は十分には確立されていない。設計の下流工程においては設計パラメータの最適化をおこなう種々の方法が提案され, 計算機による支援が行われているが, より大きな効果が期待される概念レベルの設計については, このようなシステマティックな方法は知られておらず, 全ての要求機能を満たす最適な設計解を自動的に生成することは困難である。そこで, 現実的には, 近年の設計支援研究はつぎに

示す二つの方向で進められている。第一には, 限定した判断基準のもとで計算機により自動設計を行なわせ, 提示された設計解候補の総合的な判断は設計者に委ねる方法である。第二には, あくまで設計者自体に主体性を置き, 計算機は動的に移り変わる設計フェイズを判断し, 最も適した情報を提供するために利用される方法である。

上述の第一の方法として, **遺伝的アルゴリズム** (Genetic Algorithm: GA)[15] を利用した設計解候補生成システムについて検討する。このシステムは, 設計事例から how explanation で獲得された知識を組み合わせて要求機能を達成する構造を自動生成し, 設計者に提示する。この場合, 知識の組合せ爆発の問題や, 設計解の評価を連続関数として定式化することの困難といった問題が生じるが, 本研究では GA を利用することにより, これらの問題点を解決する。

GA と従来の最適化手法との最大の相違点は, 最適化の評価関数に連続性や微分可能性などの制約が無いこと, 定義域がどのような集合であっても構わないことである。この点で GA は, 本研究の目的とする概念設計レベルにおいて, 設計解候補の生成を可能にする枠組として期待することができる。

上述の第二の方法として, 獲得された知識の質を利用可能性・合理性などの基準で整理し, 設計者の意図や動的に移り変わる設計フェーズに合わせて, その時点で設計者に有用性の高い知識を提示する形で, 設計支援するシステムの構築について検討する [16]。このとき, 設計者と計算機システムとのインタラクションが重要となる。テキスト・図・ダイヤグラムなどの多様な形態の情報が, 抽象–具体・クラス–インスタンス・全体–部分・原因–結果・統合–分解・目標–手段などの多様な形態で関係している設計知識を, 設計作業あるいは設計物の使用の際に活用することを考えると, 膨大で複雑に関連し合う知識空間の中から適切な設計知識を提示する環境が必要となる。そこで, 本研究では設計支援のためのメディア環境として, ノードに格納される情報に対して非線形かつ多様な参照形態を許容するハイパーテキスト (HYPER Text) システムのあり方について検討する [6][17]。

ハイパーテキスト環境は, 整理・統一された視点から様々な種類の情報を提供することができるため, 異なった分野, たとえば製造, 設計, 販売などに携わるメンバが, 共同で設計を行なうグループウェアとして有効であると考えられる。この環境は, 設計者から仕様者への正確で効果的な設計意図伝達のための良好なインタフェースとしても利用可能である。

このようなシステムで問題となるのは, 複雑な相互関係の中での視点の喪失である。本研究で提案する枠組では, 情報伝達過程として設計物をモデル化し, これを複雑な相互関係に秩序を与える場, すなわち設計支援情報環境の全体像とその中で現在提示されている情報の位置を示すブラウザとして位置付けることにより, この問題点を克服する。

以上のように, 本研究の目的は, 設計過程および設計対象物のモデルを構築して計算機上に記述し, 既存の設計物の解析から設計知識を自動的に抽出し, 最後に抽出された知識を様々な形態で利用する設計環境を提案し, 試作システムを構築して有用性を示すことにある。

以下, 第 2 章では, 本研究の背景として, 関連する研究, および価値工学, 公理論的設計アプローチ, 遺伝的アルゴリズム, 説明に基づく学習, 概念依存理論, 決定則, 真実性維持機構について述べる。

第 3 章では, **階層型設計知識**による設計物モデルを提案するとともに, 物理因果関係を原理的な知識とした**物理因果連鎖**による設計対象物の記述方法, および述語論理で処理できる記述形式を提案する。

第 4 章では, 知識獲得の第一の手法として, 既存の設計物が “どのようにして (how)” 目標とする機能を達成しているかの説明構造から, 概念設計支援に有効な知識を獲得する方法を提案する。

第 5 章では, 知識獲得の第二の手法として, “なぜ (why)” 一つの設計解を設計者が選んだのかを, 複数機能間の干渉回避という側面から解析する手法, および解析をシステマティックに実行する枠組を提案する。

第 6 章では, 獲得された知識の利用法として, 要求機能を達成する構造を自動生成し, 設計者に提示する形で支援する方法, および設計者の意図や設計局面に合わせて, その時点で有用度の高い知識を設計者に提示する形で設計者を支援する方法を提案する。

第 2 章

研究の背景

2.1 緒言

本章では, 本研究の背景として, 2.2節で関連する研究について述べ, 2.3節以降で理論的背景について述べる。

設計に対する計算機処理を困難にしている理由のひとつは, 現在の設計活動が設計者の経験的知識や技能に依存しており, 普遍的な設計方法論が確立されていないことにある。2.3節では, 設計方法論に関連して, 従来より機能に注目した設計物の解析および改善案の導出をシステマティックに行うグループウェアとして実践されている機能分析法 (Functional Analysis), および, 良い設計に導く普遍的な公理を仮定して設計の領域に科学的な方法をとり入れる試みである公理論的設計アプローチ (Axiomatic Approach to Design) について述べる。

また, 概念設計を対象とした場合, 数値処理可能な形で完全に対象をモデル化できないことも, 従来の数値処理能力を活用する形での計算機支援を困難にしている原因であり, 知識情報処理の導入が有効であるといわれている。2.4節では, 本研究で導入した知識情報処理手法として, 人工知能の分野で提案された説明に基づく一般化 (EBG: Explanation-Based Generalization), および AL(Artificial Life) の分野で注目を集めている遺伝的アルゴリズム (GA: Genetic Algorithm) について述べる。EBG は設計知識獲得システムを構築する基礎理論として, GA は設計知識の合成による解候補生成の基礎理論として用いることになる。

2.5節では, 設計知識表現に関連して用いることになる, 概念依存理論 (Conceptual Dependency Theory), 決定則 (Determination Rule), および真実性維持機構 (Truth Maintenace System) について述べる。

2.2 関連研究

本研究は, 知識工学的な手法を用いた概念設計に対する計算機支援方法を検討するものである。この分野に関連する研究として, 計算機を利用した設計支援を実践している CAD (Computer Aided Design), 自動的な設計を行なうための理論的モデル (設計プロセスモデル), 設計活動を一般的科学的に論じる設計理論, および自動設計だけでなく広く設計を支援するための基礎理論について述べる。

CAD

計算機援用設計 (CAD) の歴史は, 1952 年に MIT で初めての NC 工作機械が開発されたときに, NC データ生成に計算機が使用されたことから始まるといわれる。1958 年にプロッタが実用化され, 1963 年にコンピュータグラフィックスが発表されると, 設計作業の 70% を占める製図作業の合理化の手段として, CAD (Computer Aided Design) システムは自動製図 (Computer Aided Drafting, Computer Aided Drawing) システムとして発達してきた [18]。

現在の CAD システムは, 単純な製図機能だけでなく, モデリング, 図面管理, 部品形状の作成, 部品形状の登録・検索, 幾何形状のパラメトリックな変更, 部品図から組み立て図の生成など, 計算機の数値演算処理能力を活用した機能をもつに至った。さらには, 工学的解析, パラメトリック設計, 当たりチェックなど, 基本設計段階の支援に有効な機能も実現されている。

しかし, 幾何情報の取り扱いが中心的課題となっており, 設計活動の中で支援が可能な部分はごく一部に限定されている。このような反省から, 幾何情報に付加的な属性情報を加えた製品モデルの研究が進められている [19] [20] [21]。また, CAD システムが高機能化するにつれて, CAD データの欠陥や不十分さという問題が顕著になってきたため, 国際規格として 1984 年に STEP (Standard for the Exchange of Product Model Data) の開発が始まっている [22]。STEP は, すべての設計・生産段階において, 製品に関する必要な情報を完全性と整合性を失わずに表現することを目的としている。

このように, CAD の分野では, あくまで対象の構造に着目した対象モデリングを基礎として, 詳細設計段階の支援から基本設計段階の支援へと発展している。

設計プロセスモデル

設計の中でも概念設計は重要な段階である。ところが, 概念設計には本質的に困難な面が多いために, これをシステマテックに扱う方法論は確立されておらず, 個々の設計者が独自にあみ出した手法に頼っているのが実状である。これに対し, 創造過程の理解から,

設計プロセスを系統立てることのできる設計方法論を確立する研究が, 1960 年代から始められている [13]。

機械設計の領域では, 1970 年代前半から, 与えられた要求機能をいくつかの部分機能に分解することが可能なこと, および各部分機能を満足する構造を合成して全体機能を満足する機械が得られることを前提とした設計プロセスモデルの研究が続けられている。機能分解や構造合成に際しては, 機能をエネルギ, 物質, 情報の変換であるとみなして, 手掛かりとする場合が多い。

先駆的な研究として 1971 年に発表された Functional Reasoning [23] が挙げられる。これは, 部分構造には要求機能と供給機能を定義することができるという仮定の下で, 部分構造が機能の需給関係で結合されて一つの構造体を成すという考え方をベースにした設計プロセスモデルである。現在でも同じ前提に立ったモデルが提案され続けており, たとえば, 構造合成に際して冗長な構造を排除するために一つの構造で複数の機能を達成 (share) することで新奇設計を導く手法 (Function Sharing [24]) などが挙げられる。

知識工学の発展にともなって, 知識工学的手法を応用した機械系の設計プロセスモデル (たとえば [25] [26]) や概念設計に対するエキスパートシステム構築の試み (たとえば [27]) が盛んになってきた。

まず, 1970 年代後半から, 設計プロセスを, 制約条件満足問題, すなわち要求された構造的機能的関係により制約される変数へ値を割り当てる問題としてモデル化する方法が提案されている。この種の問題は制約が弱く, 領域依存知識を必要とする。

1980 年代の後半には, 様々なパラダイムに基づいた設計プロセスモデルが提案されている。たとえば, 上記の制約指向に基づく Air-cyl [28], PRIDE [29], 定性物理のもつ物理世界のモデリングの側面に着目したモデル (定性運動学 [30] など), 物理現象レベルでの知識に基づく発明的設計のモデル (類推による設計を行う EDISON [31] など), 事例ベース推論によるモデル ([32] など), 概念設計を実在事例の特徴の新たな組合せにより実現するというアイデアに基づくモデル ([33]) などが挙げられる。

また, 近年注目されている遺伝的アルゴリズム (Genetic Algorithm: GA) は, 従来の最適化手法と比べて問題表現の自由度が高い。このため, 設計問題を最適化問題としてモデル化し, GA を導入することの有効性が指摘されている [34]。

一般的な設計理論

設計プロセスのモデル化については, 様々な考え方から種々の提案がなされているが, 設計を一般的に, しかも科学的に論じようとする研究も進められており, 1970 年代の後半から, 吉川は一般設計学を展開している。一般設計学序説 [7] では, 公理論的方法が述べられている。ここでいう公理は, 統合に関する一般的方法論の一つとして提案されてお

り, 人間の概念に関する性質を公理として与えることにより, 設計過程の基礎的な性質を定理として性格づけるものである。

一般設計学が提案されたのとほぼ同じ時期の 1978 年に, Suh らは公理論的設計アプローチを提案している [35] 。Suh らによる設計公理 は, 設計を良い方向に導く原理として位置付けられる。

計算機による支援を前提としていない点で, 上記の方法論と性格が異なるが, 1970 年代初頭から, 設計物の機能に注目した分析 (機能分析) を通して, システマティックに設計改善を行うグループウェアである Functional Analysis が価値工学の分野で実践されており [36], 概念設計に有用な示唆を与えてくれる。

設計支援環境

設計プロセスモデルは, 設計エキスパートシステムにおいては推論エンジンの動作原理として位置付けることができる。これには様々な提案がなされているが, 設計エキスパートシステムの実用化のためには, 知識ベースの構築が現実的に困難であるという問題点を克服しなければならない。このような現状から, 設計知識の機械的な獲得に関する研究が近年注目され, 仮説推論, 事例ベース推論, 定性推論, 説明に基づく学習手法などの応用, またはインタビュー戦略などに基づいた手法が提案されている [37]。

設計者のパートナーとして計算機は何ができるのかを考えるとき, 自動設計だけに目を向けるのではなく, 設計者自身の設計活動をサポートする良好な支援環境を形成するものとして計算機を位置付ける研究が, ここ数年盛んに提案されている ([38][39] など)。

このように, 概念設計を対象とした知識工学の分野では, 設計プロセスモデルに関する研究と並行して, 設計エキスパートシステムを実用化するための基礎理論にも注目が払われている。

2.3 設計に関する方法論

2.3.1 価値工学における機能分析法

本節では, VE(価値工学) の分野で対象の機能構成の解明に用いられている機能分析法ついて述べた後, 機能分析に用いられる "機能系統図" について, その意義と目的, および具体的作成法を Bytheway によって導入された "**FAST** (Functional Analysis, System, Technique) ダイヤグラム" に基づいて説明する。最後に, 機能分析結果に汎用性と統一性をもたせるための方策である機能表現動詞の分類と選定の体系について述べる。

機能分析法

価値工学 (VE) とは「最低の総コストで, 必要な機能を確実に達成するため, 製品やサービスの機能分析に注ぐ組織的な努力」と定義されている [2]。VE は, 物本位の思考を捨て機能本位の思考を行うことを基本とする。なぜなら, 使用者が求めているのは機能であり, 設計物そのものや, 設計物の具体的な構造ではないからである。使用者の要求の追求が不完全であったり, 要求のすべてを把握できなければ, 以後のプロセスが完全であったとしても使用者を満足させることはできない。VE は, 一般につぎに示す 8 つのステップからなる標準手順を踏んでシステマティックに製品を分析し, 改良設計や類似の新規設計を行う方法である [40]。

1. 対象の選定
2. 原価分析
3. 機能分析
   (1) 機能の定義
   (2) 機能の整理
   (3) 機能の評価
4. 情報収集
5. アイデア発想
6. アイデア評価
7. テストと証明
8. 提案とフォローアップ

VE の最大の特徴は, 上記のステップ **3** で対象の機能を分析し, 必要な機能を明確にすることによって目的である設計物の改善を行うことにある。

(1) 機能の定義 機能の定義 [41] の意義は, 第一に要求されている機能の明確化が行われることである。

第二には, 後に行われる機能の評価を容易にすることである。機能が適切に評価されるためには, 機能を明確に定義しなければならない。

第三には, 具体的な設計物の構造から離れることによって, 思考を拡大することである。物本位の思考を捨てなければ, 機能の実現方法も現行のあるいは既成の方法にとらわれてしまう。抽象的な機能からスタートすることにより, 自由な発想を期待することができる。

(2) 機能の整理 つぎに機能の整理を行う。構成要素の各要素機能は, 相互に関連し合って作用し, 全体として対象設計物に要求されている機能を果たしている。各構成要素機能には何らかの目標があり, 同時に他の目標を達成するための手段になっている。これが, **目標 – 手段の連鎖**を形成しているのである。

目標とは, 人間が意図するものであり, この目標に基づいて人間は行動する。そして行動の結果, 目標は達成される (図 2.1(a))。この場合, 行動とは目標を達成するための手

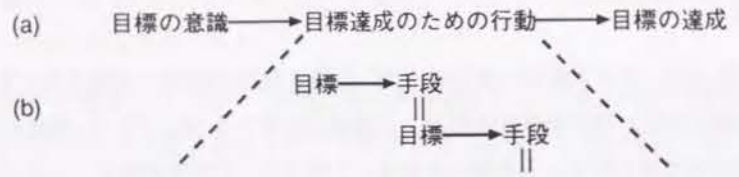


図 2.1: 目的–手段の連鎖

段を見つけ出してそれを実行することであり, もし見つけ出された手段が直ちに実行できない場合, この手段の実現が新しい目標となり, これを達成するための手段を見つけるための行動がとられる (図 2.1(b))。このように目標-手段の連鎖が成立する。この連鎖は, 実行可能な手段が見つけられるまで継続される。この考え方は, AI(人工知能) の分野で古くから目標-手段解析 (means-ends analysis) と呼ばれ, プランニング (計画作成) の最も基本的な手法の一つと考えられている。

機能整理のステップの最も重要な部分は, このような目標-手段の関係を整理して, 図.2.2 の左半面に示すような機能系統図をつくるところにある。設計物のもつ要素機能を機能系統図として整理して表現することによって, 機能間の相互関係が明確になり, より上位レベルの機能が何であるかがはっきりする。言い換えれば, その設計物にとって, 何がもっとも基本的で重要な機能であるかが明確になる。また, 機能間の重複, 干渉関係を明らかにすることができる。

機能整理の意義は, 第一に, 必要機能, 不要機能の把握が可能になることにある。すなわち, 各要素機能のうちには他のどの機能の手段にもなっていない, 言い換えれば目標をもたない機能も存在し得る。このような機能は不要であり, 具体化して実現する必要もない可能性が高い。

機能整理の第二の意義は, 機能系統図を見ることによって, 相互に関連の深い機能が集まって一つにまとまり (グループ) 単位を形成しているような機能が把握できることにある。たとえば, 図 2.2の左半面に示す機能系統図において, 機能 F2 が下位機能 F21, F22 をもち, F21 がさらに下位機能 F211, F212, F213 をもっていることから, F211, F212, F213, F22 は F2 を達成するための機能グループと考えることができる。

機能の整理の第三の意義は, 改善対象レベルが明確になることである。上位レベルの機能の改善には, 大きな改善成果を期待することができるが, 検討範囲が広くなり代替案作成のための情報収集のコストも大きくなる。下位レベルの機能の改善は, 情報収集は比較的容易であるが, 検討範囲は狭くなる。機能系統図を作成することにより対象レベルを明確にすれば, 代替案作成時の不必要な混乱を防ぐことができる。

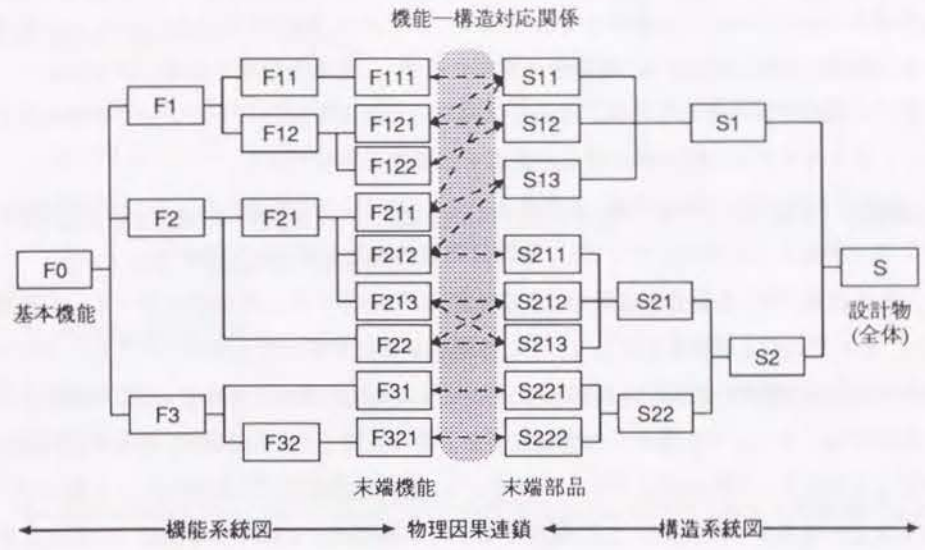


図 2.2: 機能系統図と構造系統図

機能系統図による設計物の解析

機能系統図の意義 機能系統図とは, ある機能 (基本機能) の達成を目標として設計された製品あるいはシステムに対して, その各部分構造がもつ様々な機能を適切に表現し, これらの機能に対して, 目標-手段の論理に基づいて機能間の上位-下位の関係 (目標-手段の関係) を調べ, 製品あるいはシステムの目標となる機能 (基本機能) を頂点として, その下に上位-下位関係によって機能群を位置づけたものである。すなわち, 製品あるいはシステムのもつ機能群を, その基本機能を頂点とする階層構造によって表現するダイヤグラムである。

機能を適切に表現し系統図を作成することは, その機能に対する我々の認識を新たにする効果があり, また, 今まで見落とされていた機能を再発見し, 機能間の従属, 重複, 干渉の有無を明らかにし, 改良設計をする際の指針を与える。

製品あるいはシステムは, ある特定のアイデアに基づいて設計された結果として, そのアイデアに内包された機能群をもつことになる。したがって, このようなアイデアも, 機能系統図を検討することによって明らかにすることができる。また, そのアイデアを別のものに変更することによって, 製品あるいはシステムを改良する際に不用となる機能, あるいは新しい機能と置換されるべき機能, 付加しなければならない機能などを明確に判断することができる。この様な意味において, 機能系統図は設計支援の際の有益な道具であると考られる。

機能系統図の作成　機能系統図の作成法は, VE を導入している現場でそれぞれ独自の工夫が凝らされている。その基本となる手法の一つである **FAST ダイヤグラム** への展開による機能分析法 [40][42] は, 機能を本質的に追求し, それを明確に定義できる手法として多くの実務で採用されている。この手法は以下の段階を踏むことによって半機械的に FAST ダイヤグラムと呼ばれる機能系統図を作成するものである。

機能分析法の第 1 段階では, 製品あるいはシステムの主要部分あるいは主要部品を, その部分構造としてリストアップし, 各部分構造の機能を簡潔に定義する。

第 2 段階では, 各部分構造機能 (末端機能) についてその上位機能を調べる。上位機能は, それぞれの末端機能に対して, 「この機能を果たす事は実は何をしようとしているのか」, 「この機能の出てくる原因となったさらに上位の機能は何か」, 「この機能を果たさねばならないような機能上の理由は何か」等の質問 (**上位機能質問**) に人間 (作成者) が答える事によって明らかにされる。このとき, 別の末端機能が上位機能として見いだされる場合もあるが (その場合, 上位機能となった末端機能はもはや末端機能では無い), 多くの場合, より抽象的に表現された新たな機能が上位機能として見いだされる。そして一般的には, 複数の末端機能が一つの上位機能に統合されるので, 末端機能群は幾つかのグループ (機能グループ) に分けられる。

第 3 段階では, 第 2 段階で得られた答を, 名詞と動詞の組み合わせとして「～を～する」という統一的な形式で表現し, 機能グループを作る。

第 4 段階では, 各機能グループについて, グループ内の全ての機能をリストアップし, 分析者がリストアップされた中から一つずつ機能を選び出し, それに対する「もしこの機能を果たさなくてよいとして, それでも果たさねばならない他の機能はリスト中にあるか」, 「もしこの機能を果たすとき, それにより派生する機能がリスト中にあるか」という質問 (**基本機能決定質問**) に答える事によって, 機能間の上位-下位関係を調べ, そのグループ内での最上位機能を明らかにする。そしてグループ別に, 各々の最上位機能を頂点とした系統図を作成する。

最後に第 5 段階として, 上述の方法で明らかになった各機能グループ別の系統図の頂点にある最上位機能を全てリストアップし, 基本機能決定質問によって, それらの最上位機能を明らかにし, それを最高位の機能, すなわち基本機能として, その下位に各機能グループの系統図を連結し, 全体の系統図を完成する。

機能表現動詞の選定

VE では, 「～を～する」の形式で機能を表現する。この形式を用いることによって, 対象物を「もの」としてとらえるのではなく, 対象物を使用する際に対象物がもつ働きに注目して表現することができる。VE では, 以下の点を考慮して, 日本語の動詞における

同義語・紛らわしい語が整理され, その結果として表 2.1 に示す約 140 個に限定された機能表現動詞が提案されている [44]。

1. 「～を～する」という表現において, 動詞部は必ず他動詞であり, 自動詞をすべて除外する。
2. 動詞は和語動詞, 漢語動詞, 外来語動詞の 3 つに大別することができる。機能表現用語として, 和語動詞に限定する。
3. 複合動詞は除外する。
4. 物への働きかけを表す動詞以外, すなわち, 人間および事柄への働きかけを表す動詞は, 除外する。
5. 用語の分類に際し, 用語の意味が両方に跨る場合は, 二箇所に重複して記載する。

機能系統図生成には, FAST(Functional Analysis, System, Technique) のようなシステマティックな方法も提案されているが, 基本的には全て人手に頼るものである。したがって, 同じ対象を分析しても解析結果が人によって異なることになり, 標準化の必要性が指摘されている [43]。分類・整理された用語 (表 2.1) を用いて機能を定義することは, 機能分析結果を標準化する一助となり, つぎの利点が期待できる。

- 過去の解析結果を資料にまとめるときに, 統一用語をキーにすることにより整理が容易になる。
- VE の情報処理方法としてコンピュータを導入することができる。
- 同義語・紛らわしい語の乱用による混乱を抑える。
- 同じ機能は同じ用語で定義することができる。
- 用語の解釈を統一することができる。

2.3.2 公理論的設計アプローチ

設計案を改善する方法は従来から様々な研究が行なわれているが, 大別して二つの方法がある。一つは従来から盛んに行われている, 対象システムを完全に理解し個々の要素の相互作用を把握した上で各意志決定局面に反映させる, アルゴリズム的な方法である。しかし, この方法はシステムが複雑になると莫大な資源を必要とする。もう 一つの方法は, 設計活動において常に従うべき公理が存在するという仮説をたて, 意思決定のすべての局面でこれに従うという方法である。公理論的設計法 [13][35][45][46] は, 後者の考え方に属し, 5 章および 6 章の指針となるものである。

大分類		小分類	
作用	対象物に対して働きかけを表す		
変化	対象物に対して働きかけた結果対象物そのものが変化を生ずることを表す	形状変化	対象の形が物理的に変形、分散を生じる
		状態変化	対象物が形状変化以外の物理的変化を生じる
		内容変化	対象物が化学的変化を生じ、他の異なった物質に変化する
移動	対象物に対して働きかけた結果、空間的に位置が変わることを表す	位置的移動	対象物が空間を移動する
		結合的移動	移動によって、対象物が他の物と結合する
		分離的移動	移動によって、対象物が複数に分離する

作用 18 個		当てる*	抑える	誘う	引く	打つ*	押す	示す
封じる	促す	囲む	助ける	防ぐ	補う	決める	保つ	押える
支える	吊る							
形状変化 21 個		限ぎる	縮める*	広げる	破る	切る	通す*	震わす
分ける	*砕く	均す	掘る	割る	削る	延ばす	巻く	搾る
張る*	曲げる	狭める	捻る*	乱す				
状態変化 30 個		洗う	消す*	点ける	熱する	増す	写す	搾る
照らす	練る	混ぜる	落とす*	濯ぐ	溶かす	浸す*	変える*	磨く
蓄える	慣らす	冷やす	固める	膨らます	蒸す	縮める*	煮る	乾かす
付ける*	濡らす	減らす	弛める	柔らげる				
内容変化 6 個		起こす*	変える*	消す*	焦がす	燃やす	焼く	
位置的移動 32 個		上げる	与える	浮かべる	動かす	打つ*	移す	奪う
追う	起こす*	落とす*	返す	反す	換える	交わす	下げる	沈める
捨てる	滑らす	倒す	伝える	通す*	閉じる	止める	流す	投げる
逃がす	捻る*	開く	拾う	振る	回す	導く		
結合的移動 16 個		集める	当てる*	合わせる	入れる	埋める	掛ける	重ねる
組む	刺す	継ぐ	付ける*	塗る	はめる	張る*	浸す	結ぶ
分離的移動 9 個		配る	出す	散らす	解く	取る*	抜く	除く
外す	分ける*							
複合 26 個		操る	選ぶ	覚える	書く	数える	比べる	殺す
探す	縛る	調べる	知る	刷る	作る	包む	綴じる	整える
取る*	直す	握る	計る	運ぶ	認める	見る	設ける	持つ
読む								

(* 印は複数のカテゴリーにまたがることを表す)

表 2.1: 価値工学において提案された機能表現動詞語彙

公理論的アプローチの意義

一般に設計という作業には十分な経験と豊富な知識が要求される。しかし経験豊かなエンジニアでも満足のいく良い設計は大変難しい。この原因の一つには我々を導いてくれる基本的な原理 (メタプラン) のないことが挙げられる。たとえば, 様々な設計案が存在する場合に, どちらを選べばよいかを定量的に指示するような基準が存在しない。また, 改良設計を行うときにも, どのような方向で改良を進めて行けばよいかについての基準が一般には存在しない。このような状況にあって, Suh らは設計作業・設計案を分析して普遍性のある原理をいくつか見いだし, それを設計案が従うべき公理として, 公理論的設計法を提案した。

公理論的設計法の基本的な目標は, 設計の領域に科学的な基礎を確立することにある。このアプローチは, 設計者の経験的知識, 直感, 独特なセンスや技能等に大きく依存している従来の設計作業と, 本質的に異なるものである。設計活動全般において成立し, 設計者を良い設計に導いてくれるような原理・原則が見い出されなければ, 設計はいつまでもシステマティックに扱うことができず, 設計活動というものの理解, 体系化, 教育・継承, および実践において, 解決困難な問題を残すことになる。

問題解決における公理論的手法は, 科学の歴史とともに始まったといえる。過去数世紀の間, 自然科学のあらゆる分野において, 初期の現象論的段階は, 公理や法則によって特徴付けられる, より厳密な, 現代科学の時代へと移行した。たとえば, ユークリッドは一組の公理から幾何学の体系を展開し, ニュートンはそのユークリッド幾何学に基づき, ケプラーの観測データから物体の運動や万有引力に関する法則を導きだした。また, アインシュタインが展開した公理体系はニュートンの法則をより一般化したものであり, リーマン幾何学がその基礎となっている。力学や熱力学の分野における公理論的手法による業績は, 設計の分野にも大きく貢献している。カルノー, ジュール, クラジウスらによって体系立てられた熱力学の諸法則により, たとえば, 永久機関の開発といったことは無意味であるとして検討の対象から除外されるし, 種々のエネルギ変換システムに対して効率や理論的限界を計算し評価する手段が与えらる。

同様に, 設計において何らかの公理があれば, それによって設計を概念設計段階において評価し, 本質的に悪い設計案を排除することができる。実際に設計してみないと判らないといったことがなくなり, 設計作業の能率向上が図れることになる。

しかしながら, 設計分野そのものにおいては, 力学や熱力学といった他の自然科学の分野に比べて, 公理論的手法が適用されるまでに多くの時間を費やしている。その主な原因としては, 自然現象を扱う分野のみが公理化の対象となり, 設計のような「人工的な」分野においては一般的に成立する公理体系は存在し得ないという仮定が障害になっているものと思われる。この仮定が誤りであることは, たとえば, エネルギやエントロピーと

いった物理量そのものが, それら自身直接に観測できる基本量ではなく, むしろ自然現象を説明するための人工的な概念であることからも理解することができる。

設計プロセスと設計公理

公理論的設計法では, 設計を, 機能空間における**機能的要求** (Functional Requirements: FRs) と, 物理空間における設計パラメータ (Design Parameters: DPs) との間の写像プロセスを通じて, ある対象に対する潜在的要求を満足する解 (設計解) を生成する総合 (synthesis) の過程と定義している。

機能的要求 (FRs) とは, ある設計対象に対する最小の数の互いに独立した要求項目の組である。また, 設計パラメータ (DPs) とは, 物理的実体の機能達成に関与する構造・属性を特徴付けるパラメータのことであり, 数値的なものに限らず定性的な属性記述も含む。設計公理は, この FRs から DPs への写像プロセスに適用し, 種々の意思決定局面におけるガイドラインを与えることができる。

設計プロセスの第一段階は, 設計対象物に対する潜在的要求を一組の機能的要求項目 (FRs) として定義することであり, 第二段階は, 与えられた FRs を, エネルギ・物質・情報といった形態の資源の消費を最小に抑えながら満足する物理的実体を実現するために, それら実体を特徴付ける設計パラメータ (DPs) を選択・決定することである。ここで中心的な問題となるのは, 機能空間から物理空間への写像プロセスにおいて, 「良い設計」へ導いてくれるような規則あるいは公理が果たして存在するか, ということである。もし何らかの公理が存在する場合, それを用いることによって得られる効果をまとめるとつぎのようになる。

1. シンセシスの過程における種々の決定のためのガイドラインが与えられる。
2. 種々の案の定量的な順序付けが行え, 選択が容易になる。
3. いろいろな概念やアイデアの数を制限することにより, 創造的思考に集中できる。

Suh の設計公理と系

設計公理およびそれらから派生した系は, 自然科学や数学における公理および系とほぼ同様の性質をもつ。すなわち, 定義によれば, 公理とは, それ自身を証明あるいは導出することはできないが, 反例や例外が発見されない限りにおいて一般的に正しいとされる命題である。したがって, 公理は, 対象分野におけるあらゆるケースに共通して見出される現象に着目し, 膨大な数の観察を重ねることによって初めて仮定することができる性質のものである。

系とは, 公理や他のすでに証明されている命題から直接あるいは比較的容易に導き出される命題であり, 前提とする公理や命題および導出の過程が正しい限り常に正しい。設計公理は設計活動全般, すなわち設計対象物の領域や設計過程での局面に依らず常に成立することが求められる。これに対し, 各系はそれぞれ固有の局面においてのみ有効であってもよいものとする。

独立性公理と情報量公理 Suh らが提案した設計分野における二つの設計公理を下に記す。公理 1 は機能と物理的実体との関係を取り扱うものであり, 公理 2 は設計案の複雑さを取り扱うものである。

公理 1(Independence Axiom) : 機能的要求の独立性を維持せよ

公理 2(Information Axiom) : 情報量を最小にせよ

公理 1 は, 互いに独立に定義された機能的要求が設計解においても独立性を維持されていれば, その設計は良い設計であることを保証している。換言すれば, この公理は, 機能空間中の FRs を満足させる DPs を物理空間で設定する際, その写像は, ある DP の設定を変化させたときの影響は対応する FR のみに及び, 他の FRs には干渉しないようなものでなければならないことを意味している。

一方, 公理 2 は, 公理 1 を満足する設計解のうちで情報量が最小のものが最良の設計であることを保証するものである。ここでいう情報量とは, 機能的要求を実現するために, 設計過程, 生産工程, さらには使用や保守の段階において費やされる資源 (エネルギ・物質・(狭義の) 情報) や労力の総体的な尺度である。

機能的要求を物理的実体によって実現することは, 必要とされる物理現象 (逆に言えば機能として解釈され得る物理現象) のみを生起させ, 機能とは無関係あるいは有害な物理現象の生起を排除するために, 実体に対して構造上・性質上の拘束を付加することと定義される。望まれる物理現象が生起しにくい性質のものである程, あるいは外からの干渉の影響を受けやすいものである程, 実体に課せられる拘束は厳しいものとなり, また, このような厳しい拘束条件を満足する設計物を製造するには多大な労力が必要とされる。公理 1 は, 拘束の仕方が物理現象の生起・連鎖の観点からみて適切な, 整合性が保たれたものであることを要求するものであり, 公理 2 は, 機能実現のための拘束および拘束満足のための労力を情報として捉えたときに, その量の最小化を要求するものである。

冷蔵庫の扉を例にとって二つの公理を具体的に説明する。冷蔵庫の扉を設計するときの機能的要求として,

FR1 : 外部からの熱の流入を抑え, エネルギの損失を少なくすること

FR2 : 容易に物を出し入れできること

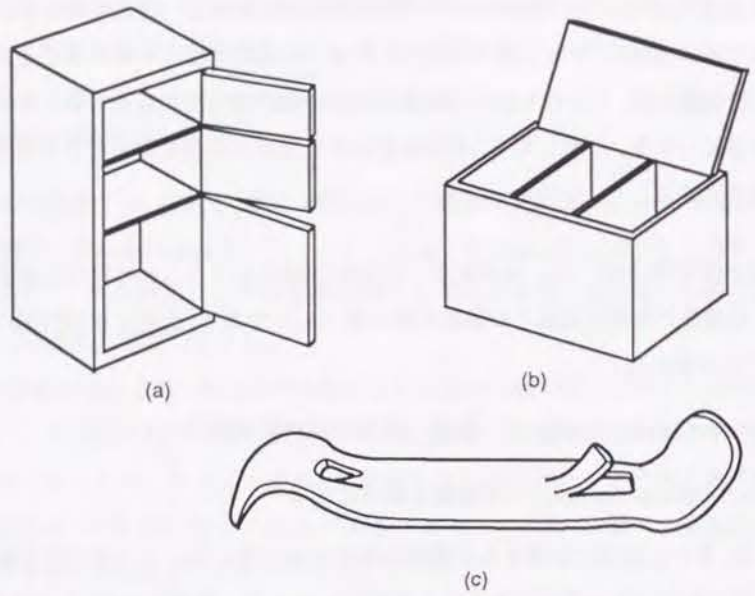


図 2.3: 冷蔵庫の扉および栓抜き兼缶切り器の例

の 2 項目を考える。家庭用として一般に見られる図 2.3(a) のような扉が側面に設けられた設計は, 公理 1 により悪い設計とされる。物の出し入れの際に冷気が流出してしまい, FR1 と FR2 が独立に満足されていないからである。このような設計を, **機能的に連結した (coupled) 設計**と呼ぶ。このタイプの冷蔵庫でも, たとえば, 冷却のためのエネルギがコスト的に重要でない場合や, 貯蔵室を区分けして複数の扉を設けた場合には, 機能間の干渉はあまり問題とならない。しかし, 扉の数を増やした設計は公理 2 に従うと情報量が多い点で良い設計とはいえない。

これに対し, 図 2.3(b) のように上面に扉が設けられた設計では, FR1 と FR2 の両方が独立に満たされてる。冷気は外気と比べて重く, 上方に流出しにくいからである。このような設計を**機能的に非連結な (uncoupled) 設計**と呼び, 機能的に連結した前者の設計と比べて良い設計である。

Suh らは機能の独立性に関し, 非連結な設計, 連結した設計の他に, **準連結な (decoupled または quasi-coupled) 設計**という, 三つ目のカテゴリーを設けている。準連結な設計とは, 設計物の使用の際, ある定められた手順に従った制御や操作によってのみ機能の独立性が保持されるような設計のことをいう。たとえば, 2 ドアの乗用車では, 後部座席の人が降車するには, まずドアを開け, 前部座席の人が降りた後, 前部座席を前に倒すという手順を踏まなければならないが, 4 ドアの乗用車の場合はこのような手続きは不必要である。乗員の乗り降りという機能に関していえば, 4 ドアの乗用車は非連結な設計であり, これに対し 2 ドアの乗用車は準連結な設計とされる。準連結な設計は連結した設

計よりは優れているが, 制御や操作の手順という「情報」の付加が必要であるという意味で, 非連結な設計よりも劣っている。

機能的連結は構造上の物理的一体化と混同してはならない。すなわち, 複数の部品を一つに統合することは, 機能的独立を維持したままで可能ならば, 情報量を減少させる (公理 2) という点で, よい設計解を生む可能性がある。たとえば, 缶切りと栓抜きが一体化した道具 (図 2.3(c)) や, 後端に消しゴムが付けられた鉛筆, ペンチ (物を挟む部分と針金などを切断する刃が付いているもの) などはいずれも機能的な連結を生じてはいない。缶切り兼栓抜き器の場合, 「缶を切る」という機能と「栓を抜く」という機能は同時に働くことは要求されていない。柄の部分は何れの機能に対しても「握られる」という機能を与えるからである。この柄の部分のように, 一つの構造が複数の機能を担うことは "function sharing" [24] と呼ばれ, 情報量を減らすための有効な手段の一つである。

公理から導かれる系　「物理的一体化は機能的要求の独立性が保たれている限りにおいて望ましい」という先に述べた知識は, 公理 1 と 2 から導き出されたものであり, 様々な設計局面において実際的な利用が可能であるという意味で系 (corollary) と呼ぶことができる。Suh らは設計公理から直接導出された系として下に記す 7 つのものを挙げている。これらはもとの公理に比べてより具体的表現で記述されており, 実際的な設計局面に直接適用することができるという意味で, より詳細な決定の支援に有効な知識であると思われる。図 2.4に二つの公理とそれらから派生した系の関係を示す。

系 1(Decoupling of Coupled Design) : 機能的要求が設計で連結あるいは相互依存している場合には, 部品や相を分割して機能的要求が互いに独立するようにせよ。

系 2(Minimization of FRs) : 機能的要求や制約条件 (Constraints: Cs) の数を最小に抑えよ。

系 3(Integration of Physical Parts) : 機能的要求の独立性が満たされるなら, 設計フィーチャ(Design Features) を 1 つの部品に統合せよ。

系 4(Use of Standardization) : 機能的要求や制約条件が満たされるなら, 標準部品や互換性のある部品を用いよ。

系 5(Use of Symmetry) : 機能的要求や制約条件が満たされるなら, 対称な形状や配置を用いよ。

系 6(Largest Tolerance) : 機能的要求を表現するときには, 許容範囲はできるだけ大きな値とせよ。

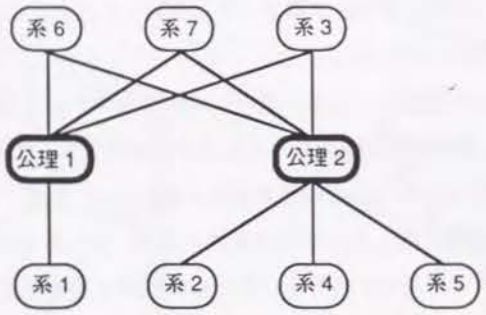


図 2.4: 設計公理と派生した系の関係

系 7(Uncoupled Design with Less Information) : 機能的要求を達成するにおいて, 連結している設計設計よりも情報量のより少ない, 非連結な設計を求めよ。

系 1 は, 設計案において FRs が連結している場合には, それらを分割して独立となるようにせねばならないことを述べている。連結した FRs の分割は, 必ずしも部品を物理的に分離することや, 新しい要素を付加することによってなされるとは限らないが, いくつかの場合においては最も有効な手段となる。

系 2 は, FRs や Cs の数が増加するほど, 設計解は複雑となり, 必要な情報量が増加することに起因している。設計物は FRs によって定義された潜在的要求を満足するものであればよく, それ以上のものでもそれ以下のものでもない。要求されるよりも多くの機能を満たそうとする設計は, 操作や保守がより困難なものとなり, 信頼性も劣る。

系 3 は, もし FRs の連結を生じないならば, 部品の数を減らすことは情報量を減じる手段として有効であることを述べている。しかし, 情報量の増加や FRs の連結を引き起こすような構造的一体化は当然好ましくない。

系 4 は常識的な設計規則である。標準部品によって FRs が実現可能である場合には, 在庫量を減らし, 加工や組立の工程に必要な情報量を減らすためにも, 特殊な部品の使用は避けるべきである。

系 5 は自明である。対称形状の部品は, 加工や組立の方向付けにおいて要求される情報量が少なくて済む。

系 6 は許容範囲の設定について述べている。許容範囲が小さいほど, 部品の加工は困難になり, 生産のために必要な情報量は多くなる。したがって, 許容差の値は FRs を満足する範囲で最大の値に設定すべきである。

系 7 は, 機能的に連結している設計よりも情報量の少ない, 機能的に連結していない設計解が必ず存在すると述べている。系 7 は公理 1 と 2 の両方からの帰結であり, この系が誤りであるならば, 公理 1, 2 とも誤りであるとせざるを得ない性質のものである。この系は, ある機能的に連結していない設計案が, 別のある連結した設計案よりも情報量が

多いような場合には, 設計の初期の段階に立ち戻って, その連結した設計案よりも情報量が少なくて, 非連結あるいは準連結な他の設計案を求めるべきであると示唆するものである。

系 7 が意味するところによれば, 公理 1 と 2 は相関関係にあり, 一つの公理命題としてまとめて表現することができるようにも思われる。すなわち, 公理 1 を満足する機能的に非連結な設計は, 連結した設計や準連結な設計に比べて常に情報量が少ないということから, 公理 1 は公理 2 の帰結であるとみなすこともできそうである [46]。しかしながら, これらの公理を二つの独立した命題として表記しておく方が, 意思決定を支援するという目的に即している。というのは, たとえば, 公理 1 がなければ, ある非連結な設計よりも情報量が少ないけれども連結した設計案を, 他の非連結でかつ情報量のより少ない解を求めることなしに, 選んでしまう可能性がある。逆に, 公理 2 がなければ, 公理 1 を満たす複数の非連結な設計案の中から最良のものを選ぶ手段がない。

以上説明した 7 つの系以外にも, さらに他の系が存在し得る。公理およびこれら 7 つの系から導かれる命題もまた系である。

2.4 学習および合成型問題解決の基礎理論

2.4.1 説明に基づく一般化 (EBG)

機械学習における一般化の枠組みは,

1. Similarity-based Generalization (類似性に基づく一般化)
2. Analysis-based Generalizaiton (解析に基づく一般化)

の二つに大別される [47]。類似性に基づく一般化は, 多数の事例からその規則性を帰納的に求めるものであるのに対し, 解析に基づく一般化は, システムのもつ領域に依存した知識を使って単一の事例を解析し, その一般化を行うものである。

Explanation-Based Learning (説明に基づく学習: EBL) は, 解析に基づく一般化をベースにした機械学習手法であり, 近年, 対象領域の知識に依存した一般化技法の研究が進められている。Explanation-Based Generalization (説明による一般化手法: EBG) は, 従来の EBL 手法を統合し, 普遍的な枠組として Mitchell らによって提案されたものである [9]。EBG 法ではつぎの四つの概念 (Terminology) が用いられる。

Goal Concept: (目標概念) 学習されるべき概念の定義。通常は操作性規範を満たさない。その意味で初期概念定義とも呼ばれる。

Training Example: (説明対象事例, 訓練例) 学習を行うために入力される正 (positive) 事例。prolog によるインプリメンテーションでは, 一連のゴール節による事

例の記述が入力情報として与えられる。

Domain Theory: (領域知識) Training Example が Goal Concept を満足するかどうかを説明し, 新たな目標概念定義を生成するために用意される, 領域固有の知識群。

Operationality Criterion: (操作性規範) システムの出力である目標概念定義が従うべき知識表現上の規範。

ここでは, ある一つのコップの表層的な記述から, コップというものの一般的な概念を学習する場合を例にとり説明する。

EBG システムはあらかじめ学習すべき概念の領域に関する十分な知識をもっていなければならない。領域知識とは, たとえば「軽くて, 取っ手が付いていれば, 持ち上げられる」といったような一般的な含意則である。領域知識によって, 入力された事例の記述から, なぜそれが目標概念の一事例であるかの説明が生成される。システムには入力として, 学習したい目標概念 (コップであること) と目標概念を満足する事例 (ある一つのコップ) の記述を与える。正事例であるコップ obj1 の構造的記述は「ジョージの持ち物である, 上面にくぼんだ部分がある, 取っ手の部分がある, 色が赤い, ...」等々の表層的なレベルでの特徴を表現したものである (図 2.5) 。このような具体的な事例から一般的な概念定義を獲得する過程を以下に示す。

システムは, まず領域知識を用いて入力事例 obj1 が入力目標概念「コップ」を満足することを説明する。ここでは上に述べた知識の他に,「底が平らであれば, 安定である」などの知識を使って図 2.6に示す説明木が生成される。この説明は, 説明木の葉の部分がすべて操作性規範を満たすまで行われる。ここで操作性規範を満たすとは, 出力知識を活用する側にとってそれが直接利用することができる形式で与えられていることを指し, その規範はあらかじめシステムに与えられる。操作性の定義については後述する。

正しい説明木が生成されたのちに, システムは, 説明木に含まれる記述のうち, 入力事例に依存し, 目標概念の充足には非本質的な情報を選択的に変数化し, 領域知識に依存する部分は定数のまま残すことにより, 説明木を一般化する (identity elimination)。この例では, handle, flat, light 等の属性は残り, obj1, han1 などの, 事例固有のアイデンティティを表現する定数は, X, P1 等に変数化される (図 2.7) 。一般化された説明木は「コップ」一般の説明を表現している。

最後に一般化された説明木の根の部分と葉の部分のみを残し, 途中の説明構造を省略したものを取り出す。これは一般的なコップの概念を満たすための十分条件を操作的な形式で記述したもの (知識) である。すなわち, 数段の木構造から成る知識をひとまとめにし (チャンク化) , 一段の知識として獲得できたことになる。操作性規範が図 2.7中の規範 (1) に設定された場合には,「取っ手の部分 (P1), 平らな底の部分 (P2), および上向

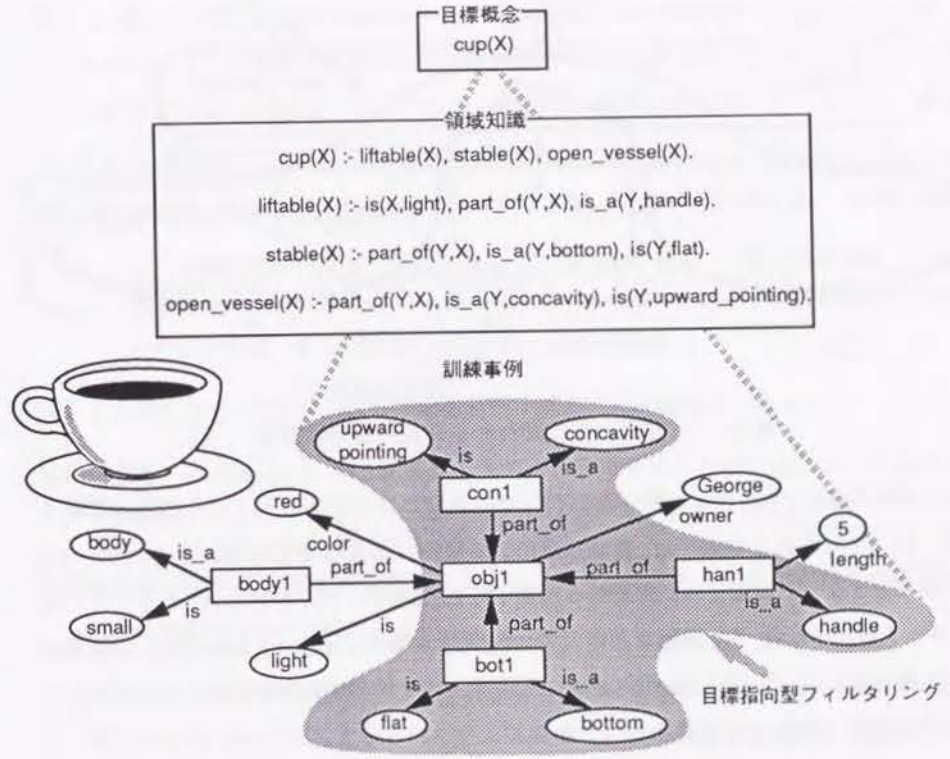


図 2.5: “cup” の例における EBG の説明プロセス

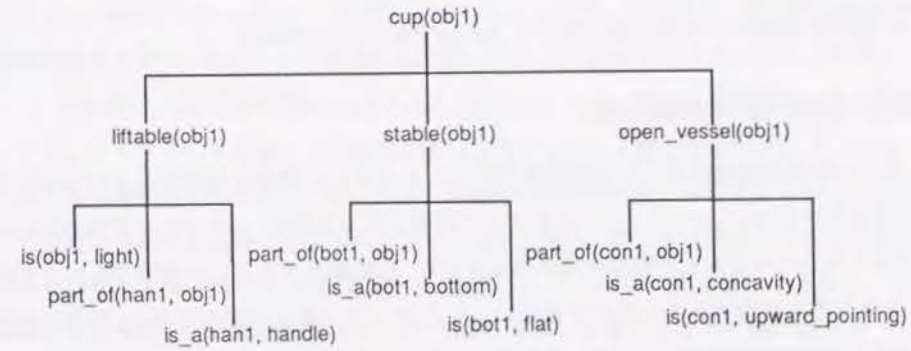


図 2.6: 訓練事例 “obj1” の説明木

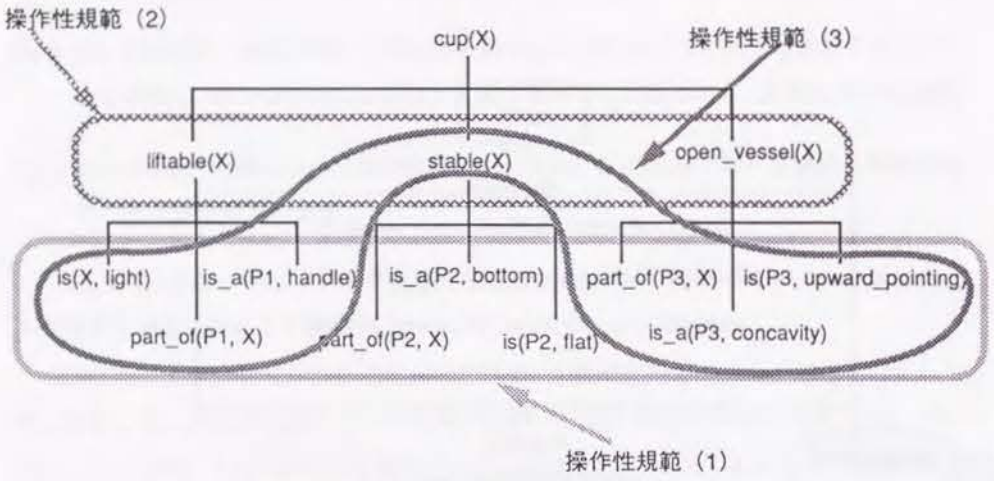


図 2.7: 一般化された説明木と 3 種の操作性規範

きのくぼみの部分 (P3) をもつ軽い物体 (X) であればコップである」という知識が獲得される。ここで得られた知識には, 領域知識による説明に必要な条件のみが含まれている。したがってコップであるために本質的な特徴 (図 2.5網掛部) だけが含まれ, そうでない特徴, たとえば, 持ち主, 色, 模様などは入力の説明対象事例 (training example) の特徴記述から除かれる (irrelevant feature elimination)。これを目標指向型フィルタリングといい, EBG 手法の特徴的な機能の一つである。

EBG 手法によって行われる一般化のタイプは, つぎの三つにまとめることができる [48]。

- 無関係な特徴の除去 (irrelevant feature elimination)
- 変数化によるアイデンティティの消去 (identity elimination)
- 操作性規範による説明木の枝刈り (operationality pruning)

操作性の定義と操作的知識の抽出

強力な領域知識に依存した演繹学習手法である EBG は, 厳密な意味で新しい知識を学習するものではない。EBG システムによって獲得される知識は, 入力された事例の記述およびあらかじめ与えられた領域知識中の記述から, 目標概念の充足に本質的なものを選択し, それらから構成されたものにすぎない。したがって, EBG 手法の意義は, 操作性規範を満足しない初期目標概念定義を, 領域知識を用いた事例の説明を通して, 操作性規範を満足する記述として新たに定義することにある。よって, 知識の操作性 (operationality) の定義および操作性規範の設定は EBG の中心的な問題の一つである。

操作性のある (operational な) 知識とは, その知識を活用する側にとって, 直接利用することができる形式で記述され, かつ, その知識を利用することによって問題解決が進展する, あるいは解決の効率が向上するような知識のことである。したがって, EBG システムがどのような種類の問題解決を支援するかによって操作性の定義は異なる。Keller は, つぎのような (概念記述の) 操作性の一般的な定義方法を提唱している [49]。すなわち, 以下の三項目

- 記述の対象となる目標概念
- 概念記述を利用して問題解決の改善を図ろうとするシステム自身
- システムが目標とする, 改善すべき効率の種類や程度

が与えられたとき, つぎの二つの要求を満たす概念記述は操作的である。

利用可能性 (usabiblity): 概念記述はシステムが利用することができるものでなければならない

有用性 (utility): 概念記述を利用することによって, システムの目標どおりの効率の改善が達成されなければならない。

再びコップの例を用いて説明する。コップの概念の知識を利用するものが, 卓上にある品物の中からコップを探そうとしている認知システムであるとしよう。知識が「持ち上げることができて, 安定したもので, 開いた容器ならばコップである」といった記述で与えられたなら, 認知システムはどの品物がコップであるか識別できない。この知識が認知という目的には漠然としすぎていて利用可能でないからであり, このような知識は操作的ではない。これに対し, 知識の記述を「取っ手が付いていて, 底が平らなもので, 上向きのくぼみがあるものがコップである」といったものにすれば (図 2.7における操作性規範 (1)), 認知システムはコップを探し出すことができる。すなわち, 認識型の問題では, 物の形状や大きさ, 色といった表層的な特徴に関する知識が有用であるからである。

これに対し, 設計型の問題の支援という目的においては, 操作性の定義は大きく異なってくる。コップとして機能する構造物を直ちに作りたいと意図する設計者にとっては, 先と同じ操作性規範 (1) によって抽出された構造的な記述の知識が有用であるが, たとえば, ある設計者が何か新奇なコップを作りたいと考えた場合, 構造的な記述よりもむしろコップの機能的な記述 (図 2.7における操作性規範 (2)) のような知識を望むであろう。また, コップの底の部分だけを改良したい (たとえば, ワイングラスのように脚の付いたコップを作りたい) と考える設計者がいたとしたら, 「安定に置ける」ことに関してのみ機能的記述にとどめ, 他の機能については構造的記述によって表現した, 図 2.7における操作性規範 (3) のような知識が有用であろう。

このように, 設計者が知識をどのような設計局面に適用し, どの段階まで設計を進展させることを意図するかによって, 操作性規範は変化する。

論理プログラミングによる EBG の実装

EBG による説明木生成プロセスは, 目標概念から一般的に成立する法則 (領域知識に含まれるルール) を用いて, 目標概念の十分条件がすべて操作性規範を満足するまで書き換えを行うものである。このプロセスの書き換えでは, 目標概念からトップダウンに演繹的な推論が行われる。これは, ホーン節論理 (Horn Clause Logic) による導出プロセスと全く同じ形態である。したがって, EBG 説明木生成エンジンを述語論理を基礎とした prolog のメタインタプリタとして作成することができる。また生成された説明木の一般化は, メタインタプリタを少し変更することで実現される。すなわち, 説明木の葉の部分 (推論木の最深部) で説明事例と単一化 (ユニファイ unify) しなければ, 領域知識に依存した引数だけが領域知識中の定数と単一化され, 事例に依存した引数は選択的に変数のまま残される。

EBG を論理プログラミングによって実現しようとする試みは, Mitchell, Keller, Kedar-Cabelli 等によって EBG が提案されて以来, Kedar-Cabelli 自身も含めて多くの研究者により発表されている [50][51]。Kedar-Cabelli, McCarty らによって提案された EBG を実現する prolog のメタインタプリタを, 図 2.8, 図 2.9, 図 2.10に沿って説明する。

純粋な prolog (pure prolog) は, 組込述語 *clause* を導入すれば, 図 2.8に示すように, prolog のメタインタプリタにより容易に実現可能である。この述語は, 第 1 引数にホーン節のヘッドを, 第 2 引数にボディ部を単一化する。ゴール節の場合, 第 2 引数は true を単一化する。

EBG においては, 入力は目標概念の正事例であるため, 説明が成功することは明らかである。それよりも, その説明のプロセスを表す説明木が重要である。通常の prolog インタプリタでは証明過程は陽に記録されず, 出力としては残らない。そこで目標概念の説明を生成する際に, 処理中のレベルよりも下位にある (より葉に近い) 証明過程の記録 (説明木) を再帰的に集め, 処理中の証明過程とつなぎ合わせて保持するための引数を第 2 引数に加える (図 2.9)。これにより目標概念全体の説明木を出力することができる。

さらに, 説明木生成と同時に一般化された説明木を生成することが可能である。証明木の葉の部分, すなわち, 推論の最深部においては, 図 2.10中の第 1 行目に示すように, 第 1・第 3 引数 (A) は, ゴール節と単一化されるが, 第 2・第 4 引数 (Ga) は, ボディ部に含まれていない。したがって, 第 2・第 4 引数は変数のままで証明木が生成されていく。


```
prove(A)      :- clause(A,true).
prove(A)      :- clause(A,B),prove(B).
prove((A,B)) :- prove(A),prove(B).
```

図 2.8: 述語 “clause” を用いた pure prolog

```
prove(A,[A])    :- clause(A,true).
prove(A,P)      :- clause(A,B),prove(B,Pb),append([A],[Pb],P).
prove((A,B),P) :- prove(A,Pa),prove(B,Pb),append(Pa,Pb,P).
```

図 2.9: 説明木 (推論の履歴) を保持した prolog

```
ebg(A,Ga,[A],[Ga]) :- clause(A,true).
ebg(A,Ga,[P],[Gp]) :- clause(Ga,Gb), copy(dt(Ga,Gb),dt(A,B)),
    ebg(B,Gb,Bp,Gbp), append([A],Bp,P), append([Ga],Gbp,Gp).
ebg((A,B),(Ga,Gb),P,Gp) :- ebg(A,Ga,Ap,Gap),
    ebg(B,Gb,Bp,Gbp), append(Ap,Bp,P), append(Gap,Gbp,Gp).
copy(Old,New) :- !,assert('$marker'(Old)),retract('$marker'(New)).
```

図 2.10: 一般化機能系統図を保持した prolog

EBG による知識獲得の問題点

一般に, 正しく演繹推論を行うためには, 完全な背景知識が必要である。ところが, 完全な背景知識からなる知識ベースを構築することは容易ではなく, 知識情報処理システム構築における大きな問題点として指摘され, 「知識獲得ボトルネック」と呼ばれている。このような現状から, 計算機による機械的な知識獲得の必要性が認識されている。

説明に基づく学習 (EBG) は, 一般的に成立することが分っている知識 (領域知識) から新たな知識を生成する手法であり, 演繹的な推論形態をとる。すなわち, EBG を用いた知識獲得を実現するには, そのための知識を獲得しておく必要があり, いわば「知識獲得のための知識獲得ボトルネック」を来たしかねない。またそこから生成される知識は, 単に既有の知識の変形であり「真の意味での新しい知識」ではないという問題が EBG の提唱以来指摘されてきた。もし逆に, 完全なる領域固有知識が整備されれば, 説明対象事例の必要がなく, 任意の知識を結合することによって新たな知識を生成することができるはずである。

4 章で提案する EBG を用いた設計知識獲得システムは, 上述の問題点を解決する機能を備えている。

2.4.2 遺伝的アルゴリズム (GA)

近年, 生物の原理を形式的に実現し工学的に応用する技術として, 人工生命 (Artificial Life: AL) が注目を集めている [52]。AL の一つに位置づけられる遺伝的アルゴリズム

(Genetic Algorithm: GA)[15] は, 生物進化の機構を形式化した計算理論・手法であり, 最適化問題や探索問題の確率的な近似解を得るのに有効な手法である。

GA と従来の最適化手法との最大の相違点は, 最適化の目的関数に連続性や微分可能性などの制約がほとんど無いこと, 定義域がどのような集合であっても構わないことである。この点で, 本研究で対象とする設計問題における解候補生成を可能にする枠組として期待することができる。計画・設計などの合成型問題に対する基本的なアプローチとして生成検査法がある。GA は先に示した目的関数定義の柔軟性に見られるように, 生成検査法として多様かつ柔軟な側面をもつ。

GA は, 生物の進化の過程における自然淘汰のシステムに着目した計算理論である。生物の進化過程においては, 遺伝子が非常に重要な役割を果たす。まず, 親の遺伝子が複製され, 子の遺伝子を生成し, その遺伝子は親と同じ性質を発現する。そのときに極めて低い確率で遺伝子の複製に誤りが混りこみ, 親とは違った形質を発現する (突然変異) こともある。こうしてできた表現型が環境への適応度という評価を経て自然淘汰され, 優秀な個体から選択的に次世代を生成することにより進化する。この進化過程は, 以下のように形式的にアルゴリズム化することができる。

1. ランダムな**遺伝子 (Genotype)** の配列をもつ個体を必要な数だけ生成し, 初期世代の集団を作る。

2. 各個体の**適応度 (Fitness)** を決める。適応度とは適用しようとする問題における個体の優秀さである。実際の生体では, 遺伝子の情報から合成された蛋白質 (**表現型 (Phenotype)**) に対して適応度の評価がなされる。遺伝的アルゴリズムでも, 同じように遺伝子から発現した表現型に対して評価を行う。

3. 適応度による期待値に応じて優秀な個体を選択し個体ペアを確率的に作成する。ペアに参加できなかった個体は淘汰される。確率的な選択の長所は, 適応度の低い個体にも選択の機会を与えることにより, 集団の多様性を維持し局所解への収束を防ぐことにある。

   設定された確率で, 任意の個体の**複製**を次世代に残す場合もある。とくに最大の適応度を示す個体を無条件に次世代に残す戦略をエリート保存戦略と呼び, 確率的な選択では適応度の高い個体でも淘汰される可能性があるという短所をカバーする。

4. 適応度に応じた確率で, 各個体ペア間の交叉 (**Crossover**) を行なう。交叉は交配とも呼ばれ, 親となる遺伝子を適当な部分で分断し, 両方の遺伝子の断片を適宜入れ換えて親の遺伝子と同じ長さの遺伝子を作ることである。

5. ある確率で**突然変異 (Mutation)** を起こす。突然変異は, 遺伝子の一部を全く別の

ものにすることで実現する。これにより, GA は集団としての多様性を維持し, 初期収束が避けられる。

以上のようにして新たな世代を作り, 新世代の個体群に対して再び同じ操作を施すサイクルが GA の概略である。世代の進行にともなって, 評価値の高い個体が発現し, 評価値が収束したとき, あるいは, ある評価値以上の遺伝子が出現したときなどが終了条件として採用されることが多い。

図 2.11に示す a から i までの全ての都市を一回づつ訪れるように最短距離で回る問題 (巡回セールスマン問題 (Travelling Salesperson Problem: TSP)) を例にした GA の応用方法を述べる [53]。TSP は NP 完全問題[1]の典型としてよく知られている。

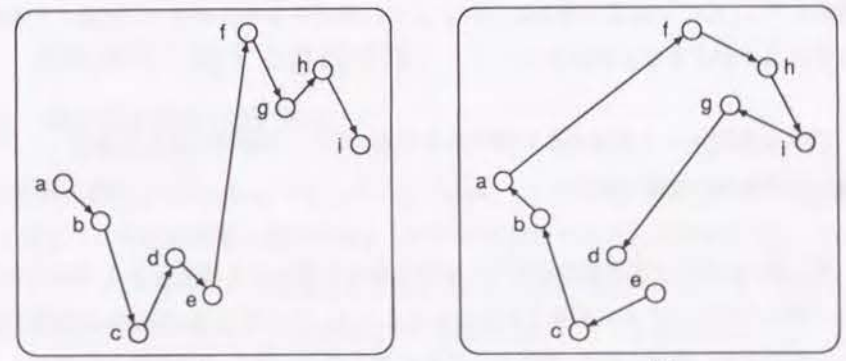


図 2.11: 巡回セールスマン問題例

遺伝子の表現 TSP 問題における表現型 (図 2.11に矢印で示す巡回ルート) を遺伝子型データとしてコーディングする。遺伝子を単純に「訪れる都市の順」でコーディングした場合, すなわち表現型をそのまま遺伝子列と考えた場合, 図 2.11(a) は [abcdefghi], 図 2.11(b) は [ecbafhigd] となる。この二つを 4 番目の遺伝子で 1 点交叉すると, [abcdfhigd] と [ecbaefghi] を得る。これらの遺伝子の表現型は, すべての場所を巡回しない。すなわち適合度は 0 になりただちに死滅する遺伝子 (致死遺伝子) である。

致死遺伝子を抑制する工夫として, つぎに示すコーディング法が考案されている [54]。まず, 都市を適当な順番に並べたリスト (以下, 都市リストと呼ぶ), たとえば [abcdefghi] を作る。表現しようとする巡回ルートを図 2.11(b) とすると, 最初の都市が都市リストの中で何番目にあるかを調べる。図 2.11(b) の始めの都市 e は都市リストの 5 番めに当たり, これを遺伝子列の最初の数字とする。つぎに, 先の e を都市リストから取り除く。つ

[1] 多項式のオーダで答えを出すのが不可能な問題。TSP では都市の数に対し指数オーダの時間がかかる。

ぎの都市 c は新たな都市リストの 3 番目にあたるので, 3 を遺伝子列の 2 番目とする。以下, 同じことを繰り返していくと, 最終的にできる遺伝子列は [532123321] となり, これが図 2.11(b) に対応する遺伝子となる。同様に考えると標準ルート (図 3.2(a)) の遺伝子は [111111111] である。

この方法でコーディングされた遺伝子に対し, 4 番目の遺伝子座で 1 点交叉するとつぎのような子を得る。この方法では, 致死遺伝子は発生しない。

親 (a): 表現型 [abcdefghi] 親 (b): 表現型 [ecbafhigd]
遺伝子列 [111111111] 遺伝子列 [532123321]

子 (1): 表現型 [abcdfhige] 子 (2): 表現型 [ecbadfghi]
遺伝子列 [111123321] 遺伝子列 [532111111]

この例のように, GA では遺伝子のコーディングに形式的な手法がなく, 対象とする問題に合わせた工夫が必要な場合が多い。

評価 TSP は最短ルートを求める問題であるため, ルートの距離の和を評価値とし, これの少ないものをよい個体とする。

交叉 適合度の大きい (評価値の小さい) 個体の中から親となる遺伝子を 2 つ選び交叉を行なう。遺伝子のコーディングを工夫することにより, この例では, 親となる遺伝子をそれぞれ一点で分断し後半 (前半) 部を互いに入れ換えるという一点交叉法を利用することができる。この例では交叉点は一つだが, 問題によっては交叉点を複数考えることもできる。交叉点が複数考えられる場合にどの交叉点で切るかは, 一般にランダムに決められる。

突然変異 一般に突然変異は遺伝子のランダムな位置のビットをランダムな値に変えることで実現するが, TSP の場合にはこの方法では致死遺伝子を数多く生成することになる。これを避ける工夫として, n 番目の数字に突然変異が起きたときは $(9-n)+1$ までの乱数を与える。たとえば, [532123321] という遺伝子の第 6 番目に突然変異が起きたとすると, $(9-6)+1=4$ より, 1〜4 の間で乱数を選び, 入れ換える。

終了条件 最後に終了条件を設ける。ここでは, 遺伝子全体の評価値の平均が遺伝子全体の中で最高の適合度をもつもの (距離の少ないもの) と一致したとき, すなわち評価が収束した場合とする。

以上で GA で処理するのに必要な事を決定した。これらの考え方を計算機上に実装し, 図 2.11に示す 9 地点を巡回する問題に対し, 個体数 60 個 で実験した結果, 評価値 100 のルートが 5 秒程度で求められた。GA を用いずに全ての組合せを探索すると, 60 秒程度で最適値 100 のルートが 18 通り得られた。

実験の結果, 最後の世代の個体の中には, 評価値 (適応度) がほぼ同じで異なるルートを示す個体が複数含まれていた。これは GA が唯一解を求めるよりも, 複数の解候補を求める問題に対して有効であり, かつそれを現実的な時間で遂行する能力をもつことを示している。すなわち, GA は厳密な最適値を必要としない問題において有効であると考えられる。本研究の対象問題としている設計, とくに概念設計においては唯一解ではなく, 設計者の要求に対して "満足できる解 (Satisfactory Candidate)"[55] を複数個生成する方が都合がよい。すなわち, この点においても GA の設計問題への導入は有効であると考える。

2.5 知識表現に関する基礎理論

2.5.1 概念依存表現 (CD 表現)

概念依存理論 (Conceptual Dependency Theory : CD 理論) は 1973 年に Schank によって提案された自然言語の意味表現および意味処理のための枠組である [5]。CD 理論の大きな特徴は, 意味表現に使用する概念を人間の心の中に存在すると予想される比較的少数の基礎的な概念の型および基礎概念への集約が行われている点である。

CD 理論によれば, 概念はつぎに示す七つの役割と六つの型に分類することができる。これを機能の表現に応用すれば, 動詞が act, 機能を供給する構造物が actor に対応する。

行為者 (actor)	行為を実行する者の概念。
行為 (act)	対象に為される行為。
対象 (object)	行為を受ける対象。
受領者 (recipient)	行為の結果としての対象を受ける者。対象の提供者も含む。
方向 (direction)	行為が指向する方向。
手段 (instrument)	行為の手段。
状態 (state)	対象の状態。
PP	名詞によって表現される物理的対象の概念の型。
(Picture Productor)	行為者, 対象, 受領者など, 様々な役割を果たす。
ACT(Act)	行為に対する概念の型。11 個の基本行為がある。
LOC(Location)	空間位置を表示する概念の型。主として方向役割を果たす。
T(Time)	絶対的または相対的な時間位置を表す概念の型。
AA(Actor Aider)	ACT を制御する概念の型 (例 : "早く", "遅く")。
PA(Picture Aider)	対象の属性を表す概念の型。 属性は状態 (値) という形式で表現される。

こうした各型の概念が, それぞれの役割に応じて図 2.12に示す統辞規則に従って相互に結合し, 概念構造を作り上げる。機能のキーワードとして用いる動詞に対応する act は CD 表現においても中核を成すものであり, Shank はつぎに示す 11 の基本行為概念 (act) を提案している。

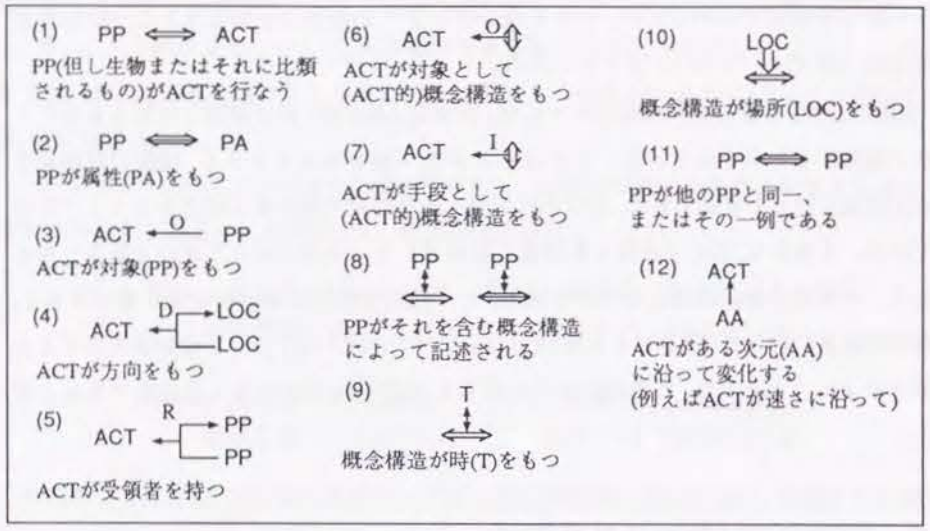


図 2.12: 概念構造を作る統辞規則

ATRANS	所有, 所有権, 制御などの, ある "物" に関する抽象的関係を移動する。
PTRANS	あるものの物理的位置を移動する。
PROPEL	あるものに物理的な力を加える。 その結果 PTRANS が起こるかどうかに関わらず用いられる。
MOVE	生物がその体の一部を動かす。
GRASP	行為者があるものを "握る" あるいは "摑む" 。
INGEST	生物があるものを自分の体内に取り入れる。
EXPEL	生物が体内からある物を外界に排出する。
MTRANS	生物の間であるいは生物内で情報を伝達または移動する。
MBUILT	生物が古い情報から新しい情報を生成する。
SPEAK	音声を発する。
ATTEND	注意を向ける。

さらに, 先の統辞規則を用いてつくり出された個々の概念に対し, 図 2.13に示す因果的概念依存関係を司る規則によって因果関係が与えられる。

図 2.14に CD 表現の例を示す。「太郎は京都に行った」「太郎が京都に向かった」など, 表層的には異なるが概念的に同一の文章は, 図 2.14(a) に示す同一の CD 表現となる。また, 「太郎が京都に着いた」という文章は, 図 2.14(b) に示す CD 表現となる。前者の文章からは, 実際に太郎が京都に着いたことが結論できない[2]ことが, CD 表現の違いとして現れている。

2.5.2 決定関係の論理的表現 (決定則)

決定則とは, 類推プロセスを正当化するために提案された知識表現である [11]。類推は, ソース事例 s において P と Q が成立するとき, ターゲット事例 t でも同じく P が成立すれば, t で Q が成立することを結論とする。

[2]我々は, 着いたことを明示的に否定されない限り着いたと推論する (default 推論)。CD 理論では, この種の推論は各基本概念ごとに設定された推論規則に基づいて行なうと仮定されている。

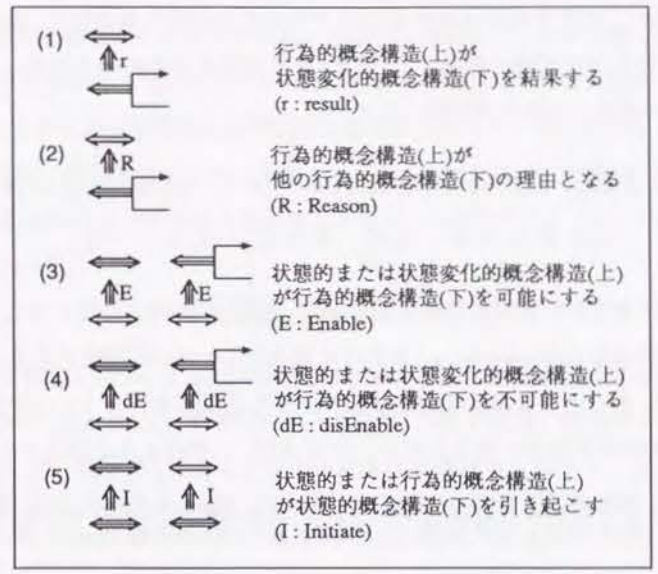


図 2.13: 因果関係的統辞規則のネットワーク表現

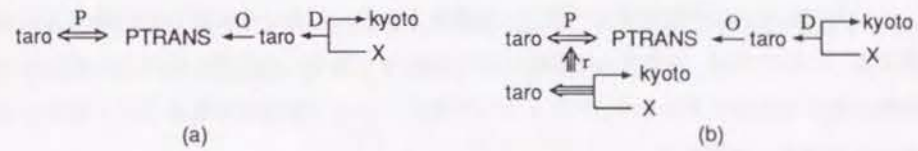


図 2.14: CD 表現例

$$\frac{P(s) \quad \wedge \quad Q(s)}{P(t)} \qquad Q(t) \quad \cdots \text{conclusion property projected from } s$$

演繹推論は通常, 十分な前提が与えられていることを前提とし, そこから論理的に正しい結論を引き出す。これに対し, 類推は, 対象とする問題領域の知識体系が十分既知でない場合においても, 知識を類似性に基づいて変換し, 試行的な推論を行なうことができるため, 学習 (概念形成) や問題解決において重要な役割を果たす。

P は, ソースとターゲットの類似性であり, 類似性に基づく推論のもっともらしさ (plausibility) の根拠となる。しかし, P がいかに多くても (もっともらしくても), 論理的に正当であるとは限らない。この類推プロセスを, つぎに示すように SIG(Single Instance Generalization) と変数の特殊化の 2 つのステップに分けて捉える。

1. $P(s) \wedge Q(s)$ から, 一般的規則 $\forall x P(x) \rightarrow Q(x)$ を結論する。
2. この一般則を T(target) によって特殊化することによって $Q(t)$ を得る。

ここで, 第 2 のステップは演繹的であるのに対して, 第 1 のステップは演繹的ではない。すなわち, 類推の正当性を失わせるのは SIG であることが明らかとなり, 類推を正当化する問題は SIG の正当化問題に帰着する。SIG を正当化するための方法の一つとして, つぎに示す知識の推論への導入が提案されている [11]。

$$\Sigma[x,y] \succ X[x,y] \ \ iff \ \ \forall y,z((\exists x \Sigma[x,y] \wedge X[x,z]) \rightarrow (\forall x \Sigma[x,y] \rightarrow X[x,z]))$$

$$\Sigma \equiv P_1 \wedge P_2 \wedge \cdots \wedge P_n \quad X \equiv Q_1 \wedge Q_2 \wedge \cdots \wedge Q_m$$

ここで, Σ, X に含まれる自由変数は, case set $\underline{x}$ (Σと X の両方に現れる), predictor set $\underline{y}$ (Σ だけに現れる), response set $\underline{z}$ (Xだけに現れる) の三つに分類される。

この知識は, どのような対象知識に対していかなる類推を行なうことが論理的に正当化され得るかの条件の宣言に他ならない。たとえば, つぎのような部品があった場合,

s1(部品 1) 形状 (円柱), 材質 (材質 1), 色 (赤), 長さ (30mm), 半径 (5mm), 弾性 (低い)
s2(部品 2) 形状 (板状), 材質 (材質 2), 色 (青), 長さ (60mm), 幅 (5mm), 弾性 (高い)

長さ 30mm 半径 5mm の円柱であり, 材質 2 でできた赤い部品 (s3) に対して, 類似性に基づいて機械的に類推すると, 部品 1 との類似点が 四つ, 部品 2 との類似点が一つであることから, 部品 3 の弾性性質は "低い" と結論される。このような誤った結論の導出を回避する一つの方法は, 通常我々が弾性性質を類推する場合には材質に注目して部品 2 から弾性は高いと結論するように, どのような対象にいかなる類推が可能かを示す知識を与えることである。本例では,

材質 $\succ$ 弾性性質

という背景知識があれば, これに保証される形で「材質 (部品 2, 材質 2) $\wedge$ 弾性 (部品 2, 高い)」からつぎの一般則「$\forall x$ 材質 (x, 材質 2) $\rightarrow$ 弾性 (x, 高い)」を生成し (Single Instance Generalization), s3 に適用して「弾性 ($s3$, 高い)」を演繹することができる。

一般に, いかなる類推が可能かを示す知識は決定関係を表すことが多い。先に示した例は "材質で弾性性質が決まる" と解釈することができる。そのため, この関係は決定則と呼ばれる。

2.5.3 真実性維持システム (TMS)

TMS(Truth Mainitenace System: 真実性維持機構) は, 一つのシステムとして捉えた知識 (命題集合) の整合性を, 知識間の依存性に基づいて非単調に維持するシステムである [14]。

従来の単調推論に基づくシステムでは, 新たな事実が加えられた場合, それまでにもつ知識からの帰結は新たな事実が加わった知識からも帰結することができる。すなわち, 事実が増えるにしたがって, そこからの帰結も単調に増加する。しかし, 通常の推論には

不完全なデータに基づくデフォルト推論, 常識推論など, 非単調な推論が伴う [56]。非単調推論では, 新たな情報がそれまでの知識と矛盾する場合には, 矛盾を導く知識が消去されるとともにその知識から導かれた (その知識の成立に依存する) 事実も消去される。そのためには, システム中の知識間の依存関係は全て記述されていなければならない。TMS では個々の命題は, 命題を信じさせる**根拠** (justification) をもつ。根拠は, 命題間の依存関係を記述するためのラベルである**節点名**とともに, つぎに示すような一つのノード (節点) として表される。

[節点名] : [$justification$]

ノードは, *In* **状態** (*In* state) と *Out* **状態** (*Out* state) の二つの状態の何れかをとり, 何れの状態であるかは justification に記述された (他の) 命題の状態に依存する。

justification は, 以下の (1) あるいは (2) のような意味をもつリストとして定義される。なお, 以下の [In リスト], [Out リスト] はそれぞれ節点名のリストであり, [結果] は節点名である。

(1) 支持リスト正当化 (support list justification : *SL*) [In リスト] に含まれている全てのノードが In 状態にあり, [Out リスト] 内の全てのノードが Out 状態にあるときに限って, 当該ノードの状態が In になる。

(SL [In リスト][Out リスト])

支持リスト正当化はつぎの 3 種類に分類することができる。

1. [In リスト] = [Out リスト] = $\emptyset$ の場合, 常に In 状態となるため, このような正当化をもつ命題は**前提** (premise) と呼ばれる。
2. [In リスト] $\neq \emptyset$ かつ [Out リスト] = $\emptyset$ の場合, In 状態のノードが増加しても当該ノードの状態が In から Out に変わることがないため, 単調性をもつといえる。
3. [out リスト] $\neq \emptyset$ の場合, In 状態のノードが増加すると当該ノードの状態が In から Out に変わり得る。 すなわち当該ノードは, 暫定的に In 状態にある**仮定** (assumption) を表すと考えることができるため, 仮定ノードと呼ばれる。

(2) 条件つき正当化 (conditional proof justification : *CP*) [In リスト] 内の全てのノードの状態が全て In, [Out リスト] 内のノードの状態が全て Out という条件のもとで [結果] に記述されているノードの状態が In になるときに限って, 当該ノードの状態が In になる。

(CP [結果][In リスト][Out リスト])

TMS は, 上で定義したような命題とその状態集合を保持しており, 新しい justification が入力されるとその妥当性の判定し, もし妥当であるのにこれにより正当化されるノードの現在の状態が Out である (矛盾が生じた) ときには, それを In にするとともに, それによって波及する他の信念状態への影響を, つぎに示す手順を繰り返すことによってチェックし, 相互に矛盾のない状態をつくり出す。

手順 1 矛盾発生の原因として, 矛盾ノードの foundations (*In* リストの直接的および間接的根拠を与えるノード) の中の仮定ノード (ただし他の仮定ノードの foundations になっているものを除く) の組合せを探す。

手順 2 この組合せが矛盾発生の原因であることを CP-just. により明記し, 同時にこの CP-just. と等価な SL-just. を求める。

手順 3 矛盾の原因となったノードのうちの一つを「犯人 (culprit)」と見立て, それが暫定的に信じられていた (In になっていた) ことが誤りであると考え, その根拠 (SL-just.) の outlist のノードの一つを Out から In に変更する。

手順 4 In に変えられたノードに対して, inlist として矛盾ノードおよび culprit 以外の仮定ノードからなるリスト, outlist として culprit の just. の outlist の中で状態が Out から変わらないノードからなるリストを, 新たな根拠づけ (SL-just.) として与える。

手順 5 culprit を Out に変更するために In に変えられたノードに依存する他のノードの状態を, 依存関係を辿りながら変更する。

手順 6 手順 5 に伴う新たな矛盾が発生した場合, 1 から繰り返す。

TMS が, 時々刻々変化する状況下で正しい知識の維持を目指し, 一つの選択枝のみを活性化し, それを深く探索して矛盾を回避する縦型探索を行なうのに対し, すべての選択枝を活性化しておき, 矛盾が生じたとき, その矛盾を示す枝を打切り, 残りの矛盾を生じていない選択枝を並行的に探索 (横型探索) する Assumption based TMS (ATMS) が提案されている [57]。

ATMS は, 複数の解を同時に比較することができるため, 最良解を求める必要のあるシステムに適用することができる。また, 大局的状態の概念をもつため, 注目する選択枝の変更が可能である。 このような点で ATMS は TMS の問題点を克服している。一方 TMS では, 推論が冗長になったり首尾一貫しなくなるという ATMS のもつ問題は生じない。

例として, 「鋼またはゴムを棒状または螺旋状にした部品で衝撃吸収機能を達成したい」との条件の下で TMS の動きを示す。TMS はまず表 2.2におけるノード N1 ～ N4 を生成する。たとえば, ノード N1 は材質がゴムでないなら鋼であることを SL just. によって記述し, 暫定的に In 状態にある仮定を表す。

節点	意味	just.	状態	just.	状態	just.	状態
N1	材質 (鋼)	sl([],[N2])	in		out		in
N2	材質 (ゴム)		out	sl([N6,N3],[])	in		out
N3	形状 (棒)	sl([],[N4])	in				out
N4	形状 (螺旋)		out			sl([N8],[])	in
N5	矛盾 (弾性)	sl([N1,N3],[])	in		out		
N6	nogood(N1,N3)	cp(N5,[N1,N3],[])	in				
N7	矛盾 (高温)			sl([N2],[])	in		out
N8	nogood(N3)			cp(N7,[N3],[])	in		

表 2.2: TMS の例題

いま, 「棒状の鋼では目的機能が達成できない」ことを TMS に伝えると, 矛盾ノード N5 が生成される。TMS はこの矛盾発生の原因を手順 1 によって探索し, 本例の場合 N1, N3 を見い出す。この N1, N3 の組合せが不都合であることを (N1, N3 から 矛盾 N5 が導かれること) を CP-just. をもつ nogood ノード N6 によって明記する (手順 2)。

つぎに TMS は, 仮定 N1, N3 のうちの一つを矛盾発生の犯人 (culprit) と見なし, それが暫定的に In であったことがことが誤りであると考え, その根拠 (SL-just.) の outlist の一つを Out から In に変えるとともに, 新たな根拠づけを手順 4 に従って与える。たとえば N1 を犯人として選ぶときには, 表 2.2の中列に示すように, N2 を In に変え, $sl([N6, N3][])$ を新たな justification とする。N2 が In に変わったことに伴って N1, N5 が Out に変わる。

さらにもし, 「部品の使用環境が高温下であるため材料としてゴムが使えない」との情報が得られると, 矛盾ノード N7 が生成され, TMS は矛盾の犯人を見つけて nogood ノード N8 に明記する (手順 2)。犯人は唯一 N3 であり, その SL-just. の中の outlist は唯一 N4 である。結局, 表 2.2の右列に示すように, N4 は新たな根拠とともに In となり, それにともなって N3, N2, N7 が Out, N1 が In となる。

最終的なノードの状態は, 螺旋状 (N4) の鋼 (N1) を使った部品ですべての矛盾を回避しつつ目的の機能を達成することを表す。

2.6 結論

本論文は, 概念設計に注目した, 設計知識の表現・獲得・利用方法の提案で構成されており, 本章では, このために必要とされる基礎理論について述べた。

設計知識の表現を考えたとき, 概念設計を対象とする場合には表層的な構造情報だけでなく, その背後にあり構造を規定する情報を直接操作できる必要がある。本研究では,

3 章において 2.3 節で述べた機能分析法に基づいた設計プロセスモデルおよび設計対象物モデルを提案する。また, 2.5 節で述べた概念依存理論に準じた機能表現の解析によって, 設計知識の計算機内記述に統一性を与える。

設計知識の獲得については, 4 章において提案する設計知識獲得手法は, 2.4 節で述べた説明に基づく学習法の枠組を拡張するとともに, 2.5 節で述べた決定則を導入して実現する。また, 5 章において提案する設計知識獲得手法は, 実在設計物に対する 2.3 節で述べた公理論的設計アプローチの視点からの機能–構造関係の解析に基づく。2.5 節で述べた真理維持機構を利用して, この解析をシステマティックに実現することができる。

設計知識の利用については, 6 章において 二つの方法を提案する。その内, 特定の評価尺度から概念設計解候補を生成する手法は, 一般に優良な設計であることの評価は定式化が困難である, 全ての解候補を現実的な時間内に調べつくすことが不可能である, 等の問題を本質的に有しており, 2.4 節で述べた遺伝的アルゴリズムの導入が有効な対象であるといえる。

第 3 章

設計知識表現

3.1 緒言

本章では, 概念設計支援のための設計活動モデル, およびそれに基づく設計対象物モデルを提案する。

1 章で述べたように, CAD の分野では, 幾何情報のみならず様々な構造属性をも含めた形での製品モデルが種々提案されている [19][20]。しかし, これらはあくまで対象の構造にのみ着目したものであり, たとえば次世代 CAD データの国際標準である STEP [22] においても, 属性による構造データに主眼が置かれている。

このような設計対象モデルでは, 対象物の構造に対する処理は可能であるが, 構造を規定する機能および機能と構造を関連づける原理・原則に対する情報処理という, 設計の本質的な部分を扱う概念設計段階を対象とする場合には不十分であると考えられる。

そこで, 本研究では, 従来より機能に着目した設計物の解析を行なっている価値工学 (Value Engineering: VE) の視点を導入して, 物理因果関係を原理・原則とし, 機能概念を陽に表現する設計対象モデルを提案する。

以下, 3.2節では, 一般に人間の選択行為が従うべき三つの合理性 (目標合理性・因果合理性・経済合理性) を導入して, 合理性に基づく継続的な選択行為として設計活動をモデル化する。さらに, このモデルに基づいて, 基本機能・機能・構造属性・実体に関する知識と, 各々を媒介する三つの合理性に対応した知識 (プラン, 物理因果関係, 実装化知識) が階層を成す階層型設計知識空間, およびこれによって記述される設計対象物モデルを提案する。

3.3節では, 測定領域を例にとり, 目標・プラン・機能・物理因果関係・構造属性からなる五つの設計知識のそれぞれについて述べる。

3.4節では, 機能に対して計算機上での表現および操作が可能な記述形式を与える。機能の表現は, 価値工学の分野の表現法である「～を～する」という動詞とその目的語の表現形式を採用する。このような抽象度の高い形式を採用した場合, 機能の意味内容が一義

には定まらないことから, 機械的な取り扱いが困難となる。したがって, それぞれの挙動概念の意味を明確にする必要が生じる。そこで, VE において機能分析の統一化・標準化を目的として提案された約 140 個の機能表現動詞に対して, 概念依存理論に基づいた意味分析を行ない, 機能を表現する動詞語彙を選定した。これによって, 機能表現動詞に込められた意味的内容 (概念構造) が明らかになるため, 計算機上での記述, ならびに解析結果の標準化が可能となる。

3.5節では認識公理に基づいて, 部分構造 (構造素と呼ぶ) とその関係, および個々の構造の属性によって, 客観的に設計物の構造を表現する形式を提案する。この形式は, 意味ネットワークで表現可能であり, 性質・属性継承などの推論機能をもつ。

3.2 合理性に基づく設計活動の分析

本研究で提唱する設計知識表現法を説明するための事例として, 特許申請された設計物である溶融鉛の局所流速測定装置 “可撓 (とう) 棒 (Flexible Rod)”[58] を用いる。図 3.1に, 可撓棒の外観と, 意味ネットワークを用いてコーディングした表層的情報 (構造特徴) を示す。

高温流体の局所流速測定装置「可撓棒」

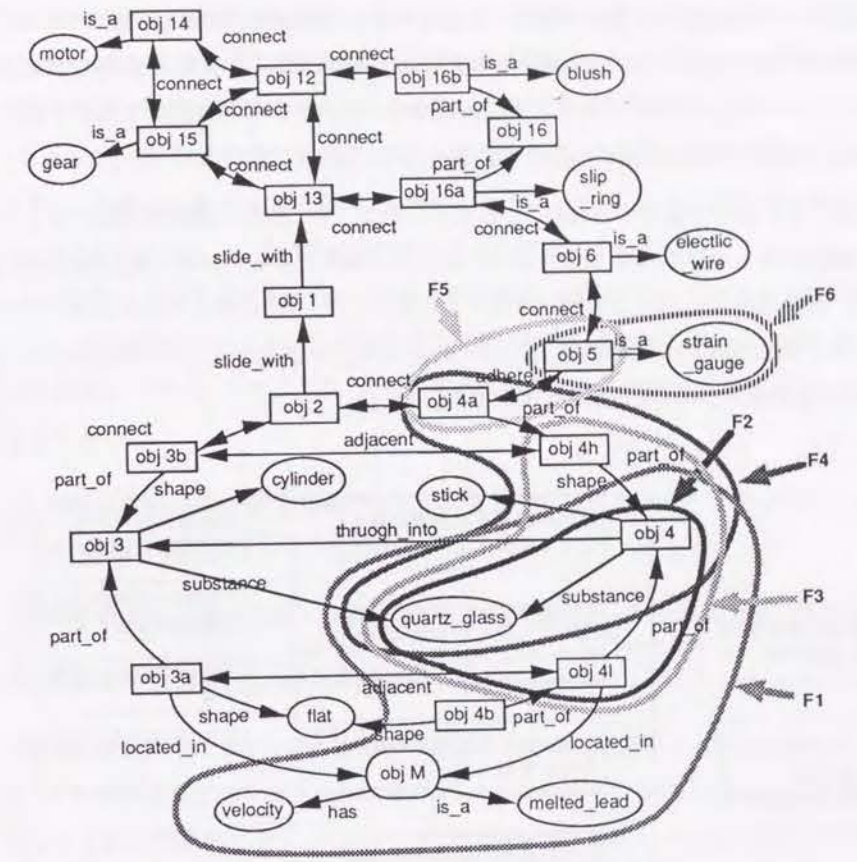
【特許公報 (B2)7027-2F】

動力によって回転する垂直回転軸に沿って上下位置調整可能に, 左右方向ガイドに沿って左右方向に位置調整可能に支持腕を設け, この支持腕に耐熱材の筒管を支持せしめ, この筒管を貫通する耐熱性の可撓棒の上端は前記支持腕に保持せしめ, この可撓棒の下端は前記筒管下端により適宜, 突出せしめてなり, この可撓棒の前記保持箇所近辺にはストレーンゲージが貼布され, このストレーンゲージの出力を接続する接続手段を前記垂直回転軸とこれを支承する本体との間に設けてなる高温流体の流速測定装置

上記の文章 (特許クレーム) は, 特許請求の範囲を定義するものであり, 特許権者の権利範囲を拡大するために, できるだけ一般性をもたせた言葉を用いて記述されたものである。

可撓棒の測定対象は槽 (11) の中を流れる溶融金属 (M) である。弾性体からなる可撓棒 (4) の一端 (4a) は, 支持腕 (2) で支承され, 片持ち梁となっている。溶融金属に浸された自由端 (4b) はスプーン状の形状をしており溶融鉛の速度に関連した流体抵抗を受け, 可撓棒をたわませる。弾性体である可撓棒には, たわみに関連した表面引張応力が発生し, これを歪ゲージ (5) で電気的出力に変換するものである。

このようなブランレベルの説明によって, 特許権者のアイデアをはっきりさせることができる。電磁流速計, 羽根車の回転による流速計, ピトー管やオリフィスを使用した流



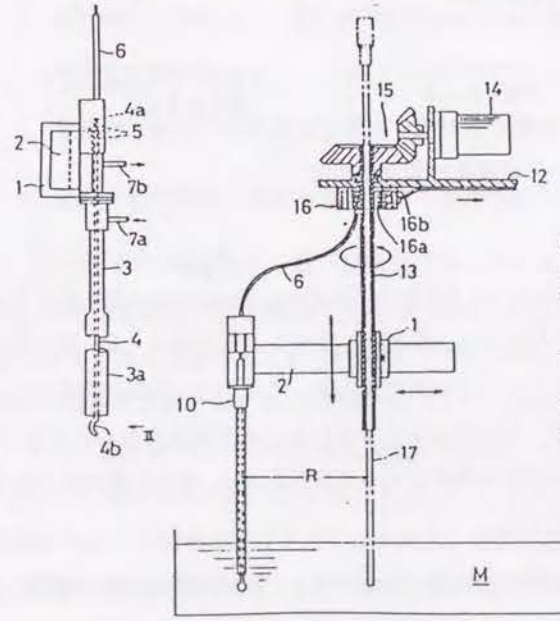


図 3.1: 可撓棒の構造

速計などの一般的な測定プランと際だって異なる点は, 弾性体の表面に発生した応力による物理量の伝達である。これは, 測定対象が高温であることによって, 一般的な測定プラン (方式) では生じる解消不可能な干渉を回避するために特許権者が採用したアイデアと考えられる (弾性体である限り, 高温であることは, 応力に影響が及ばない) 。

価値工学における機能分析法は, 2 章で示したように, 機能系統図を作成することによって設計物のもつ目標-手段の関係を明らかにする手法である。本例の機能系統図は, 基本機能 “高温流体の局所流速を測定する” を頂点として, 図 3.2のような木構造として与えられる。機能系統図の末端機能 (図 3.2 (F1)〜(F6)) を達成するための構造特徴は, 図 3.1における部分ネット (破線部) に対応する。

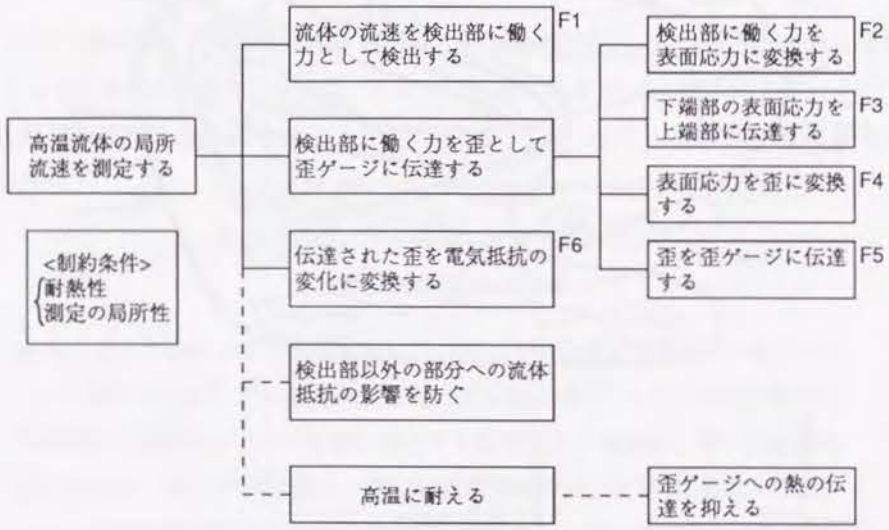


図 3.2: 可撓棒の機能系統図

機能系統図作成には, FAST(functional Analysis, System, Technique) [59] のようなシステマティックな方法も提案されているが, 基本的には全て人手に頼るため, 機能系統図が機能分析法としての汎用性を失うという問題が生じる。すなわち, 各部分構造のもつ機能 (末端機能) あるいは上位機能が, 系統図の作成者自身によって定義されることは, 作成者によって同一の機能が異なった言葉で定義される可能性を生み, 機能定義の仕方次第で機能分析の結果も異なったものとなる。

機能の定義だけでなく, 機能間の関係の決定においても, 上位機能質問, 基本機能決定質問のように, 人間との対話が重要なステップであることが, 系統図作成に対して作成者の主観が大きく影響し, 作成者によって異なった系統図が得られる原因となる。

また, 機能系統図の作成手順の初期の段階で “部品” の機能が定義されたのちにボトムアップに機能系統図が作成されることは, 機能系統図作成者の思考が即物的になり, なかなかアイデアの想起につながらないという問題点もある。

これらの問題点に対して, 本研究では, 対象物に含まれる構造・機能・目標 (基本機能) 群の間に, 合理性に基づいた知識を設定することによって, 人間の直感のみに頼るのではなく, システマティックな分析を可能にすることを考える [4]。

人間はある目標を達成するために, 様々な行動の中から理性に従って適当な手段を選び出す。設計活動においても, 様々な局面で人間は多くの設計案の中から合理的に有効なものを選択してゆく。このような, 人間の選択の基礎となる合理性として, つぎの三つが考えられる [3]。

- **目標合理性** 一定の価値基準のもとで設定された目標の達成のために選択した手段が, 目標との関連の意味で合理的か否か
- **因果合理性** 手段の選択と目標達成の間の因果関係が, 合理的に把握できるか否か
- **経済合理性** 素材, コスト等の選択が合理的か否か

設計活動をこのような合理性に従って目標とその手段を選択する活動であると捉えるとき, その結果実現される個々の設計物は, これらの合理性に特徴付けられる合理性知識を内包した具体的事例となり, 以下のように解釈 (説明) づけることができる。

1. 設計物の目標に対し, それを達成するために必要な機能群を目標合理性に従って選択する。
2. 各機能達成のために必要な構造・属性群を因果合理性に従って選択する。
3. 選択された構造・属性を実現するための手段を経済合理性に従って選択する。

したがって, 設計物を一般に説明付けるためには, 目標集合空間 (G), 機能知識空間 (F), 構造知識空間 (S), 実体集合空間 (R) の導入が必要である。設計物は, その目標機能 ($g \in G$) とそれを達成するための機能群 ($f \subset F$), 各々の機能を支える構造属性群 ($s \subset S$) と各々の構造を実現するための実体群 ($r \subset R$) から構成される。

1 つの設計物がこれらの知識集合空間の要素から構成されていることを, 人間の直感に頼らず, 機械的に説明するためには, N, F, S, R 各々の間に, これらの関連付けとその根拠を与える知識群を設定する必要がある。まず, 目標集合空間と機能知識空間との間には, 目標合理性に基づいて, ある目標をそれを達成するための一連の機能に展開するプラン型知識群 (プラン型知識空間 (P)) の導入が必要である。また, 機能知識空間と構造知識空間の間には, いわゆる物理法則知識群 (物理因果法則空間 (L)) を設定することによっ

て, ある機能の達成に対し, ある構造が必要とされている理由を因果合理性に裏付けられた形で示すことができる。構造知識空間と実体集合空間の間には, 一般的法則が前もって設定できず, 経験的知識に頼らざるを得ないが, 形式的には経済合理性に基づいた実装化知識空間 (I) を考えることができる。

以上の考察から, 図 3.3に示すように, 目標 (基本機能)(G)・機能 (F)・構造属性 (S)・実体 (R) の 4 空間と, 隣接する空間を継ぐプラン型知識空間 (P)・物理因果法則空間 (L)・実装化知識空間 (I) の 3 媒介知識, 計七つの空間から成る知識を用いて設計物を解釈することができる。

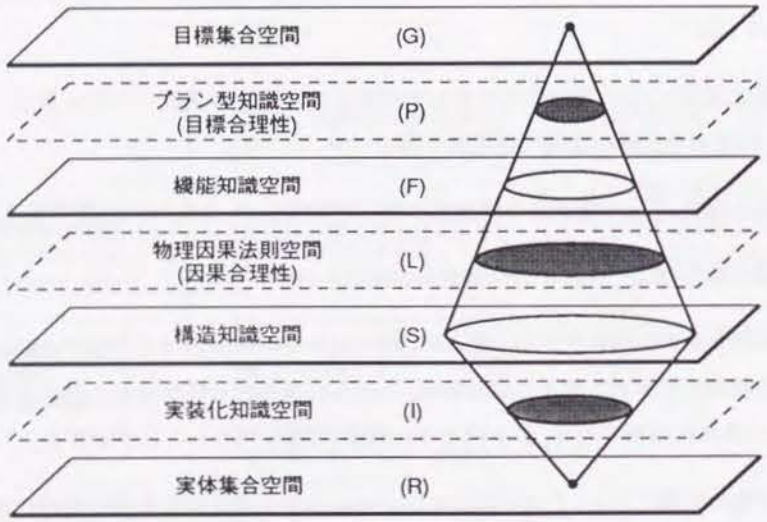


図 3.3: 階層的設計知識空間

3.3 階層的設計知識

本節では, 3.2節で示した, 設計物の基本機能の成り立ちを表現するための七つの知識空間 (**設計領域知識**) のそれぞれについて考察する。本研究では概念設計に焦点を当てているため, 実体集合空間 (R), および実装化知識空間 (I) については考えない。以下に測定器領域を対象として, G, P, F, L, S の各々に対応する設計領域知識を示す。

3.3.1 目標集合空間 (G)

本章の例として取り上げた測定機器の目標は, 測定対象のもつ物理量を測定することである。測定対象のクラス, 物理量のクラスに応じて, 目標が特定されるとともに, それに付随した目標, たとえば耐熱性, 耐食性, 測定範囲の局所性などが, この空間内にある。

3.3.2 プラン型知識空間 (P)

プランとは, 設計物の目標 (基本機能) を達成するために, この目標を一連の機能 (副目標) に展開する知識を表すものである。したがって, 設計者が設計物に盛り込んだアイデア, ひらめきなどは, 主にこの空間内に顕著に現れるものと考えられる。

一般に, 測定機器領域には様々なプランが存在するが, それを予め用意しておくのは事実上不可能である。しかし, 測定のためには何れにせよ, 測定対象の測定したい物理量を測定器の一部分の物理量として取り込み (detect), 変換器 (transducer) まで伝達 (transmit) し, 変換器によって, 所定の物理量に変換する, という一連の機能の遂行が必要となる。したがって, このプラン型知識空間において, 測定プランは, 「取り込み→伝達→変換」という機能列で記述される。

3.3.3 機能知識空間 (F)

機能知識空間内の全ての機能は, 物理法則や物理効果を介して, 構造知識空間内の構造属性と関連づけられる。したがって, 各々の機能に対しては, 「機能が果たされるためにはそこに物理因果関係が成立している」という最も一般的な知識が, 予め与えられる。

被測定物からの検出という機能や, 変換器における変換機能は, どちらも物理量の変換によって達成される。検出器から変換器への物理量の伝達機能のように, 複数の伝達を結合して 一つの伝達と考え得るような機能は, 再帰的な定義によって表現される。また, 伝達の途中にも物理量の変換が行われる。

すでに図 3.2で示した可撓棒の機能系統図のように, 機能系統図は, 目標集合空間 (G) 中の目標 (基本機能) を, 媒介知識であるプラン型知識空間 (P) を介して機能知識空間 (F) へ展開したものである。

3.3.4 物理因果法則空間 (L)

物理因果法則の一般的な表現として, 図 3.4に示す二つの型を考えることができる。

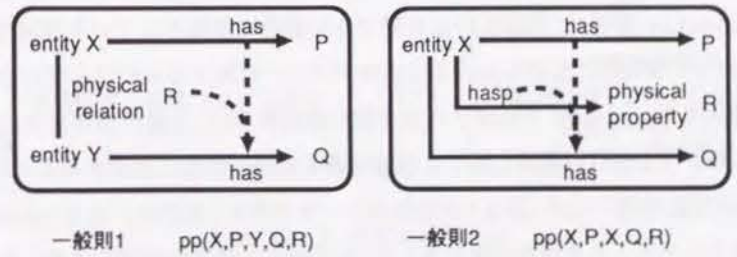


図 3.4: 物理因果関係の一般形

一般則 1　実体 X と Y の間にある物理的関係 R が存在するとき, 実体 X のもつ物理量 (またはその変化) P と実体 Y のもつ物理量 (またはその変化) Q の間にある因果関係が成立。

一般則 2　実体 X がもつ物理特性 R が存在するとき, 実体 X のもつ物理量 (またはその変化) P と Q の間にある因果関係が成立。

一般則 1 は, 2 つの構造物にどのような関係があるとき, 各々の構造物のもつ物理量 (またはその変化) の間に因果関係が成立するかを規定する知識であり, たとえば, 「流体の中に固体が挿入されていれば, 流体の流速と固体の曲がりの大きさに因果関係がある」というものである。

一般則 2 は, ある構造がどのような性質をもつ場合に, その構造内でどのような因果関係が成立し得るかを規定する知識であり, たとえば, 「弾性体 (elastic) ならば, そのたわみ (bend) と表面引張り応力 (tension) との間に因果関係がある」というものである。

この一般則は, 含まれる変数 X, P, Y, Q, R が具体的対象に特殊化されて, 実際の物理因果関係の表現となる。たとえば, 一般則 2 の変数 X と Y を弾性体に, P と Q を応力に, R を物理的結合関係に特殊化すれば, 「力の伝達」の表現となる。

可撓棒の測定プロセスは, 図 3.5(a) に示すような法則群の連結した因果連鎖として説明される。すなわち, 力の発生法則 L1 における液体 M を溶融鉛 (melted lead) に特殊化し, 歪伝達効果 L5 における弾性体 Z1 を歪ゲージに特殊化する。これによって, L1 から L6 までの法則が連結されて, 一つの因果連鎖 (測定の道筋) が形成される。

3.3.5　構造知識空間 (S)

機器に含まれる構造・属性群は, 意味ネットワークを用いて構造化して表現する。たとえば, 可撓棒の構造要素に, 石英ガラス棒, 石英円筒ガラス, 歪ゲージ, 溶融鉛などがあるが, これらに関する意味ネットワークの一部は, 図 3.6のようになる。構造・属性を表すために使用する用語は, 統一的なものを用意しておく。たとえば, 部品と部品の接続関係は connect という用語で表され, 物理的属性をもつ場合は hasp (has property) で表される。is_a, part_of 関係は, 構造設計の分野でも一般的に使用されている表現である。

is_a を使って階層的に記述された下位のカテゴリーに属するものは上位のカテゴリーの属性を継承する。part_of で記述された下位の構成要素は, 上位の構成要素の属性を継承する。このような属性の継承に関する推論知識をこの空間内に用意することにより, たとえば, 解析対象事例の記述 (図 3.6中破線のアークで表す) の中に「部品 (object3) の材質が石英ガラスである」という情報があれば, あらかじめ構造知識空間に蓄えられている情報 (図 3.6中実線のアークで表す) から「object3 は透光性をもつ」ことを推論することができる。

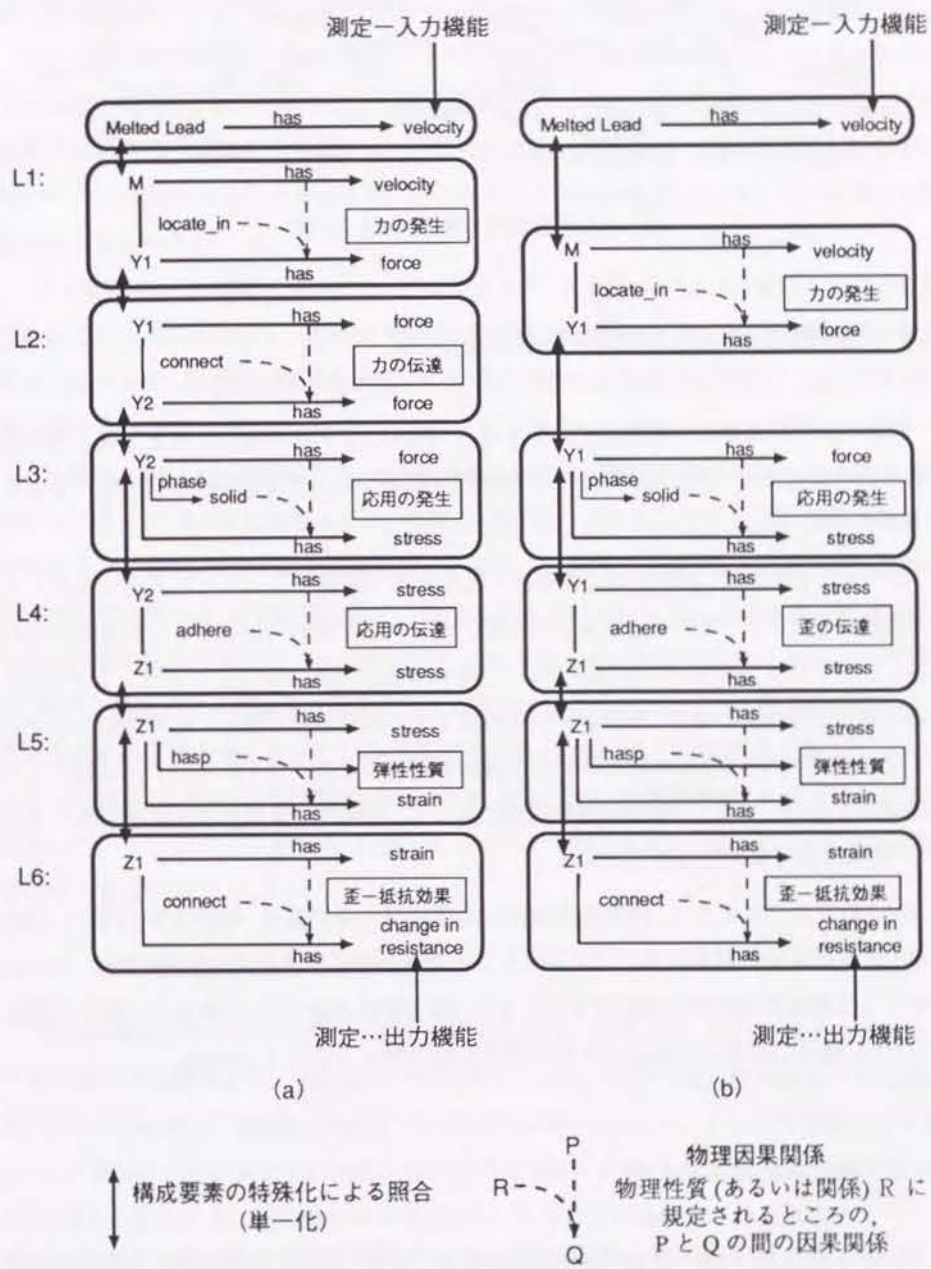


図 3.5: 可撓棒の因果連鎖

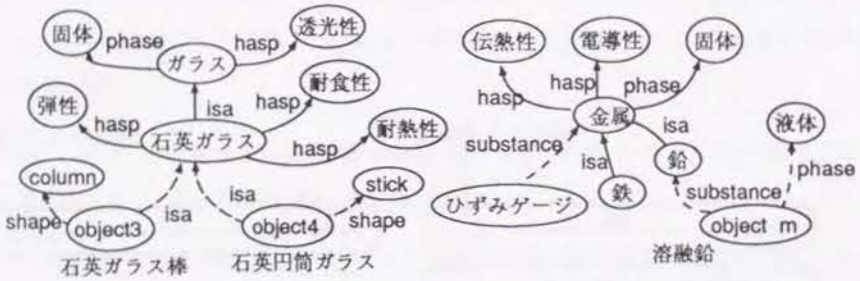


図 3.6: 構造知識空間内知識 (一部)

3.4 機械的推論のための機能表現形式

機能概念を計算機上で機械的に操作するためには, まずその表現方法を定義する必要がある。本研究では, 価値工学の分野の表現方法に基づいて「[目的語] を [動詞] する」という形式を用いる。

このような形式は, 設計方法論の分野に見られる「物質やエネルギの入出力関係」という機能の捉えかた, または定性推論の分野に見られる「特定の目的をもった視点からの挙動」という捉え方に較べて制限が少なく, 柔軟な表現形式であるといえる。しかし, この表現形式を採用した場合, 動詞および目的語となる名詞を計算機上に処理可能な形で記述する必要が生じるが, 日本語だけで数千語ある動詞のそれぞれを計算機上に定義するのは現実的でない。そこで, 機能表現動詞語に関して, 意味的な重複や包含関係を明らかにして, 語彙空間を整理する必要がある。

2.3.1節で述べたように, 機能表現動詞の整理に関する問題は, 価値工学 (VE) の分野においても機能分析結果の統一化を目的として追求されている [44]。本研究では, これを参考にして機能表現動詞を分類するとともに, 概念依存理論 (CD 理論 [5]) に準じた解析を行ない, 個々の動詞間の意味的重複や包含関係を明らかにする [60][6]。

3.4.1 機能表現動詞の分類

すでに, 表 2.1に価値工学 (VE) の分野において, 機能分析の実務経験から得られた規則に従って分類整理された約 140 の機能表現動詞語彙を示した。しかし, その中には物理現象では説明できない動詞 (示す, 助ける, 補うなど) や, 目的論的な視点をもち, 客観的・標準的な動詞として相応しくない動詞 (誘うなど) も含まれる。本研究では, 物理構造物を対象とするので, 力と変位を表す動詞 (作用, 形状変化, 位置的移動) に注目する。

3.4.2 機能表現動詞の概念依存表現

各々の機能表現動詞の意味内容を明らかにし, 計算機上での表現および操作が可能な機能記述を定義するために, CD 理論 (conceptual dependency theory) [5] を導入する。

2.5.1節で述べたように, CD 理論の最大の特徴は, 人間の心の中に存在すると予想される小数の基礎的な概念の型および基礎概念を用いて, 自然言語文の意味表現に使用する概念を表現 (CD 表現) する点にある。表層的には異なるが意味内容は同一である文章は, 同一の CD 表現が与えられる。また, 陽に示されていない意味内容[1]も, CD 表現では明示的に与えられる。

この特徴は, CD 理論においては, 読み替えなど, 構文論的手法では困難な自然言語処理を目的として利用される。本研究では, 機能の概念構造を統一的な書式で図式的に表現することにより, 機能間の概念的上位下位関係を機械的に判断する目的で, CD 理論の特徴を利用する。

上述のように, CD 理論は元来, より広範囲な自然言語処理に用いられるべく構成されている。とくに, 11 個の基本 act は, 生物 (人間) の行為を表現するために選択・定義されており, 物理的設計物の機能表現には不必要な act がある。逆に, 自然言語理解においては重要ではないが, 物理的構造物を表現するためには明確にする必要のある概念, たとえば対象物の移動経路に関する情報などを CD 理論では簡潔に表現できない。したがって, 物理世界における対象物の因果関係を効率よく表現するためには, 概念の要素・型を定義し直す必要がある。本研究では, VE で提案された表 2.1 の分類体系に準じて, 図 3.7 のように表 2.1の動詞群の基礎的な概念 (作用, 形状変化, 位置的移動) を表現する形式を定義した。たとえば, 「S が O を A から B に移す」という記述は, 図 3.7に「位置的移動」で示す CD 表現をもつ。

3.4.3 機能表現動詞語彙の階層構造化

言語の意味分析といった極めて複雑な問題を対象とする以上, 試行錯誤的なアプローチを採用する必要がある。機能表現で明らかなことは, 物理的構造物の機能となる動詞が, 動作主 (actor) と対象物 (object) と対象物の属性 (attribute) という下位構造をもち, actor と object の型は共に CD 表現における PP(名詞によって表現される物理的対象の概念) であり, 動詞が ACT (行為に対する概念) であることである。これを基礎として「対象物を位置的に変化させる動詞」を概念依存構造表現した結果, 140 の動詞はつぎの点から区別できることがわかった。

1. 対象物の属性が, どの役割に分類されどの型をもつか。

[1] 2.5.1節で用いた例における「太郎が実際に京都に着いたこと」など。

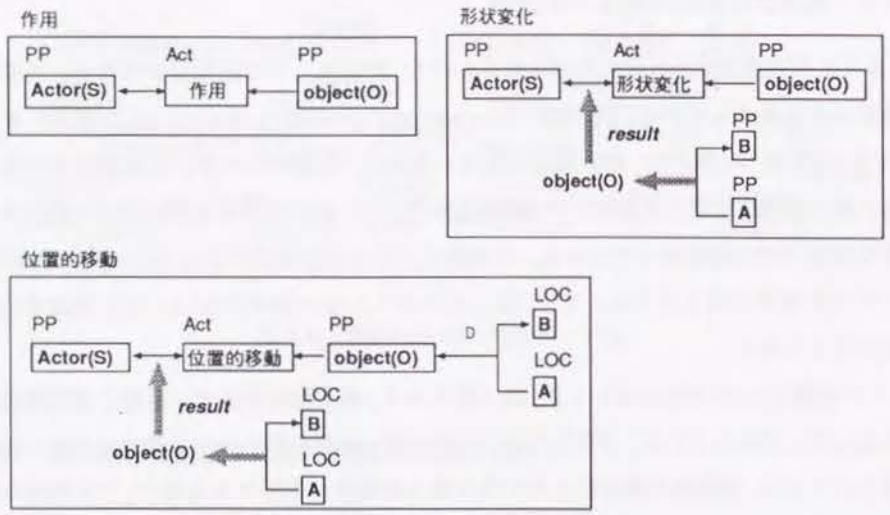


図 3.7: 物理設計物の機能表現動詞の CD 表現

2. actor や object にどのような条件が付与されているか。

3. 動詞が「属性を変化させるもの」の場合, 変化の経路が存在するが, その変化の経路 (path) に, どのような条件が付与されているか。

たとえば,「通す」という動詞と「流す」という動詞について考えると,「通す」の方が「流す」より一般性をもち, その区別は「流す」という動詞の目的物には "流体である" という条件が付与されていることである。このような関係によって「物理的構造物の機能表現動詞」を分類する。

その結果から, 動詞群を図 3.8に示すように階層状に整理した。「物理的構造物の機能となる動詞」は, その概念構造に動作主 (actor), 対象物 (object), 対象の属性 (attribute) をもち, actor と object は PP という型であるとの条件が付いている。その下位概念に,「対象物の属性を変化させる動詞」があり, これは対象物の属性に initial attribute(i_a) と final attribute(f_a) があるという条件が付与されたものである。さらにその下位概念に「対象物の属性を位置的に変化させる動詞」があり, これは, (i_a) と (f_a) が位置 (LOC (空間位置を表現する概念)) という型をもつという条件が付与されたものである。リンクに付いているラベルは, mfy(modify) が下位構造に条件を付ける等の修正を加えること, dif(differentiate) が下位構造の条件を変えて区別することを表す。

図 3.9は,「対象の属性を位置的に変化させる動詞」に分類される動詞を, 概念的上下関係によってさらに詳細に階層構造化した結果であり, 図上方が動詞の概念的上位を表し, 動詞間のリンクは動詞のどの下位構造に条件が付与されたのかを示す。

図 3.8: 機能表現動詞の階層化 (一部)

$mfy3$(path) というラベルをつけたリンクは,「移す」に対して「伝える」は, 対象物の移動経路が特定されている場合に使う動詞であるという点で概念的に異なることを表す。「伝える」以下の動詞は「経路を特定された位置的移動」を表すことになる。「A から B にガスを流す」,「A から B へガスを通す」,「A から B へガスを導く」が同じ機能を表すことは,「流す」,「通す」および「導く」が全て「経路を特定された位置的移動」という概念の下位に位置することに対応する。動詞の階層構造化によって, 概念的に下位にある動詞のそれぞれを定義する必要はなく, 上位の動詞の定義をその下位の動詞に継承させることが可能になる。

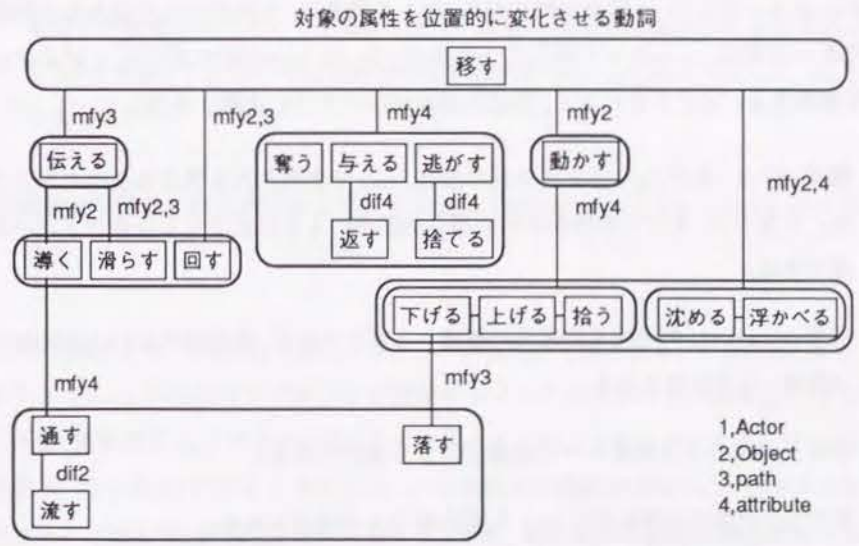


図 3.9: 位置的移動に分類される機能表現動詞の階層化

3.5 認識公理に基づく構造表現形式

一般設計学によれば, 実体は属性によって認識あるいは記述することが可能である (認識公理)[7]。属性概念とは, 無数の性質を有する個々の独立のインスタンスがもつ性質の中から抽象した共通な性質である。したがって, 実体は属性によって認識あるいは記述することが可能であり, また属性項目を増やせば増やすほど, 実体は正確に記述されると考えられる。

このような, 属性の意味付けあるいは解釈は, 人間が行うことであり, 設計属性の定義は人間の対象理解及び認識に依存し, 人間の価値概念が反映されることになる。すなわち, 設計目標に応じて主観的に設計属性が選ばれることになり, 客観的で汎用性がある設計属性の数え上げは現実的には不可能である。しかしながら, 設計の計算機支援を考えたときには, 属性記述と工学的意味との一義的な対応が必要である。すなわち, 同一の属性値をもつならば, そのときに限り工学的意味で同一な実体でなければならない。したがって, 現実的な計算機システム構築を考慮すると, あらかじめ設計対象の範囲を限定しておき, なおかつその範囲の中で設計属性項目を適切に設定する必要がある。これによって, 実体は属性によって記号化され, 計算機処理が可能となる。

このような, 一般設計学に見られる "実体" の概念は, " 設計対象物の物理的構造" であると考えられる。一般に設計対象物はさまざまな構造の単位が複雑に関係し合い, 組合わさって成り立っている。まず, 設計対象物を物理的に構成する単位を「構造素 (s)」と呼ぶ。また, 同じ構造素から成るものでもその構造素の組み合わせ方によって全体は異なったものになる。そこで,「構造素間の関係 (r)」を導入し, それぞれの役割をもつ構造素 (s) が, 互いに関係 (r) し合って成り立つ構造 (S) を, 図 3.10に示すように, r および s を用いて表現する。以上をまとめて, 構造記述を以下のように定義する [8]。

- 構造 (S) は, 実体 ($s_i(a)$) と実体間の関係 (r_{ij}) で表現される構造物全体の記述である。すなわち, すべての構造単位の複合的関係のもとに成立する全体およびその関係である。
- 実体 ($s_i(a)$) は, 対象物を物理的に構成する単位であり, 実体ラベル (s_i) と実体のもつ属性 (a) で表現される。
- 実体ラベル (s_i) は実体ラベル語彙 ($\mathcal{S}$) から選択される。
- 属性 (a) は数組の属性項目 (a_i) と属性値 (v_i) で表現される。
- 属性項目 (a_i) は属性項目語彙 ($\mathcal{A}$) から, 属性値 (v_i) は属性値語彙 ($\mathcal{V}$) から選ばれる。
- 継承・包含関係 ($is\_a \in \mathcal{A}$) は, 実体ラベル (s_i) を引数にもつ特別な属性項目である。

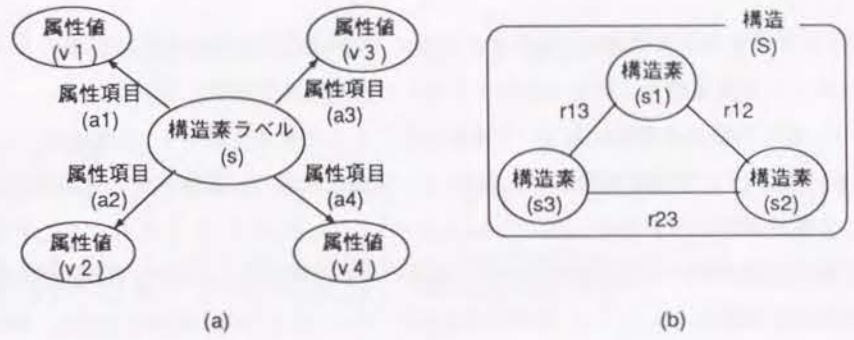


図 3.10: 意味ネットワークによる構造の記述形式

- 実体間の関係 (r_{ij}) は, 関係記述子語彙 ($\mathcal{R}$) から選択される。

$\mathcal{A}$(属性項目語彙) 色・材質・形状などの単純な項目から, 継承・包含関係 (is_a) のような性質継承による推論を可能にする項目までを含む。

$\mathcal{V}$(属性値語彙) 棒状 (rod), 板状 (panel), 箱状 (box), 椀状 (spoon) ··· 等の属性値である。

$\mathcal{S}$(実体ラベル語彙) 認識公理に従えば, 実体はそれのもつ属性により意味づけされる。したがって実体ラベルそのものには意味はなく, 実体ラベル語彙には $obj(n)$ のような識別子が準備されるだけである。ただし, 構造と機能の関係が明らかなもの (「部品 (p)」と表す) については通常用いられる名称 (spring, gear, nut, ···, etc.) を割り当てる。

$\mathcal{R}$(関係記述子語彙) 物理設計物を構成する実体 (物理的なまとまりつ部品) 間の全ての関係。

この定義により, 構造 (S) は意味ネットワーク表現が可能となる。すなわち, 図 3.10(a) に示すように, 1 つの実体ラベル (s_i) を表現するノードとそれのもつ属性値 (v_j) を表すノードを, 属性項目 (a_j) でラベル付けされたアークで結ぶことによって構造素 ($s_i(a)$) が表現され, 図 3.10(b) に示すように s_i と s_j を実体間の関係記述子 (r_{ij}) でラベル付けされたアークで結んで, 構造 (S) が表現される。また, 幾何形状と機能の関係が明らかなもの (部品 (p)) を表す実体ラベル ($p_i \in S$) に, 通常用いられている部品名を割り当てることにより, 計算機システムだけでなくシステムの操作者にとっても理解が容易になる。図 3.1 に示した可撓棒の構造は, この表現形式に従うものである。

3.6 結論

設計を支援する計算機環境を構築するために, 設計過程および物理的設計物のモデル, およびそれらを計算機上に記述し操作するための表現形式を提案した。

まず, 設計活動を合理性に基づいた選択行為であるとする立場から, 目標機能 (G), プラン (P), 機能 (F), 物理因果関係 (L), 構造 (S), 実装化知識 (I), 実体 (R) の連続的選択行為として設計過程をモデル化した。これに基づいて, G, P, F, L, S, I, R に対応した階層型設計領域知識を用いて表現される設計対象モデルを提案するとともに, 各々の設計領域知識の内容を検討した。とくに, 物理因果関係については 2 つの一般形を提案し, 物理現象に支配される設計物を物理因果関係の連鎖 (物理因果連鎖) という統一的な形での記述が可能なことを, 例題を用いて示した。

提案した階層型設計領域知識は, 統一的な形式をもち, かつ設計の深層的な部分である, 機能および機能と構造を連関させる物理因果連鎖を陽に表現するものであり, 概念設計段階を対象とした設計支援に有効であると考えられる。

機能の表現に関しては, 機能表現動詞に対して CD 理論に準じた解析を行なって意味内容 (概念依存表現) を明らかにすることにより, 動詞間に意味的重複や包含関係を与え階層構造状に整理する方法を提案し, 位置的移動を表す動詞を例にとって説明した。

構造の表現に関しては, 設計対象物の物理的構造は属性によって認識あるいは記述することが可能であり, 属性項目を増やすほど正確に記述することができるという, 一般設計学の分野で提案された公理に基づいて, いくつかの属性ラベル (a) と属性値 (v) の組によって記述された構造素 (s) と, 構造素間を 1 対 1 に結ぶ関係 (r) を用いてネットワーク状に表現する方法を提案した。構造の記述は 4 章以降で提案する知識獲得および利用システムに対する入力となるものであり, システム操作者の入力データ作成の労力を軽減する意味でも, また構造表現に機能的内容を含めると機能推論の本質的な意味を失うという指摘 (定性推論における NFIS (no-function-in-structure) 原理) に沿う意味でも, 本章で提案した表層的かつ客観的な構造記述は妥当であると考えられる。

第 4 章

設計物からの設計知識獲得

4.1 緒言

近年, 設計の支援に有効な知識情報処理手法が種々提案されている。しかし, それらを実装した計算機システム, いわゆる設計エキスパートシステムは, プロトタイプの域を越えていない。1 章で述べたように, その理由の一つとして, 設計知識を書き下して設計知識ベースを構築することの困難, すなわちエキスパートシステム構築における “知識獲得ボトルネック” が指摘されており, 専門家 (エキスパート) と知識エンジニアによる手作業の知識獲得に替わり, 計算機によって支援された半機械的な知識獲得の必要性が認識されている。

本章では, 既存の設計物の表層的構造情報を解析し, その設計物が内包する機能達成のための種々のレベルの設計知識を抽出する手法を提案する。この手法は, 説明に基づく一般化 (Explanation-Based Generalization : EBG) 法 [9] を導入し, 設計事例において, その基本機能が下位の機能群にどのように展開されているか, 各末端機能がどのような物理因果連鎖により達成されているか, さらに, 各物理現象がどのような構造・属性に基づいているか, の説明づけをトップダウンに行う。この説明の履歴から, 機能・物理現象・構造の各々のレベルにおける設計知識を一般的・操作的な形で抽出する。

以下, 4.2節では, EBG 手法を, 3 章で示した 5 つの空間からなる設計物のモデルに適用する。この場合, EBG における説明木は, 設計物の機能・構造がどのような継がりをもって基本機能を達成しているかを説明するものとなり, 一種の機能系統図であると捉えることができる。

4.3節では, 論理プログラミング環境下で計算機上に実装した機能系統図生成およびその一般化のための推論エンジン, さらに, 5 つの空間からなる設計領域知識について述べる。また, EBG の問題点を明らかにする。

4.4節では, 操作性規範, すなわち「獲得された知識は予め与えられている領域知識に対してどのような意味で新たな (操作性が加わった) 知識といえるか」を定める規範が予

め固定されているという EBG の問題点を解決する一つの方法として, 単一事例から種々のレベルの設計知識を獲得する方法について述べる。

4.5節では, 予め与えられた領域知識を演繹的に組み合わせた知識が真の意味で「新たに獲得された知識」といえるのかという EBG のいま一つの問題点の解決法として, 設計領域知識の表現形式を, "A ならば B" という含意命題の形式から "A が B を決める" という知識 (決定則) [11] [61] に拡張した場合の知識獲得手法について述べる。

4.6節では, 4.4節, 4.5節で提案した手法によって獲得される設計知識の質 (操作性) を, 適合性・合理性・効果性の 3 つの側面から総合的に評価する。

4.2 設計に対する説明に基づく学習手法の導入

設計型の知識情報処理システム, とくに機械設計を対象としたシステムを構築するためには, 機械的に設計知識ベースを構築することが重要であると認識されており, 仮説推論・事例ベース推論・定性推論・説明に基づく学習手法などの応用, またはインタビュー戦略など, 各種の設計知識獲得手法が提案されている [37]。本研究では, 以下に示す理由により, 2.4.1節で述べた領域知識説明に基づく学習手法を応用した設計知識獲得システムを構築する。

機能系統図に表現された機能構成や機能と構造の結び付きに関する知識は, 設計物の改良や概念設計に有益な情報を提供する。しかし, つぎに示す二つの問題点のために, この情報はそのままでは概念設計のための普遍的・一般的な知識表現を与えるものとはいえない。

第一の問題点は, 分析結果が特定の設計物に固有のものになっていることである。ある設計物に対して作成した機能系統図を, そのまま機械的に他の設計物に適用することはできない。しかし, 一度行った分析結果は, 適切に一般化され類似設計物に対して適用可能にすることが望まれる。

第二の問題点は, 機能系統図が一意的には決定しないことである。すなわち, 機能分析の方法や分析を行う人に依存して, 上位機能と下位機能とのつながりについての考え方や機能を定義する用語の不統一性から, できあがった系統図も異なったものとなる。これは, FAST などの系統的な手法を採ることによってある程度統一をはかることができるとしても, 基本的には人手によるものであり, 十分に統一化されているとはいえないことによる。

このような問題点に対して, 3 章では人間の直感的判断に頼らずに, 構造・属性群, 機能群, 目標 (群)(基本機能) と, これらの間を結びつける合理性を反映した知識であるプラン型知識, 物理法則知識を設定するとともに, 概念依存理論と認識公理に基づく設計知識表現形式を提案した。このような知識を用いて機能分析することにより, 統一的に定

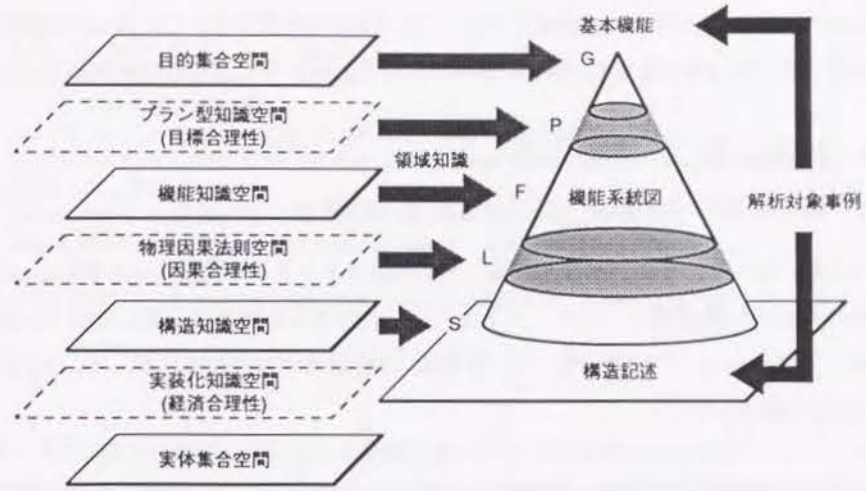


図 4.1: 階層型設計知識空間と EBG の対応

義された機能・構造表現からなる機能系統図を作成することができる。さらに本章では, EBG(Explanation Based Generalization : 説明に基づく一般化) 手法を導入して, 一意的な機能系統図を自動的に生成するとともに, 生成された機能系統図を一般化し, そこから一般的に利用可能な形で設計知識を獲得する方法について検討する [4][10][12][62]。

EBG 手法を, 五つの空間からなる設計物のモデルに適用した場合, 目標集合空間 (G) に対して目標概念 (Goal Concept) を, 構造属性集合空間 (S) の一部に対して説明対象事例 (Training Example) を対応付けることができる。このような対応を取るとき, 目標集合空間からプラン形知識空間 (P), 機能集合空間 (F), 物理因果法則空間 (L) を経て構造属性集合空間に至るそれぞれの空間上にある事実あるいは規則 (一般的・法則的知識) を, 目標概念と説明対象事例とを結び付ける領域知識 (Domain Theory) として利用することができる。その結果生成される説明木 (Explanation Tree) は, 設計物の部分機能, 構造がどのように基本機能を達成しているかを説明するものとなり, 一種の機能系統図であると捉えることができる。

EBG によって機械的に生成された機能系統図は, 3 章で提案した設計知識語彙に統一されている。また, EBG の一般化能力により, 個々の事例に制限されない一般的な機能系統図を得ることができる。

図 4.1に, 提案するシステムの構成を示す。システムは, 解析対象事例の基本機能, および, 表層的な構造を入力すると, あらかじめ保有している五つの空間からなる設計知識を用いて, なぜそのような構造で基本機能が達成できるかについての説明木, すなわち機能系統図を生成する。つぎに機能系統図を, 解析対象事例に限定されない形に一般化す

る。一般化された機能系統図を以下では一般機能系統図 (GFD: Generalized Functional Diagram) と呼ぶ。GFD の部分木に注目し, それをチャンク化することによって獲得される記述は, 統一的な書式をもち, 解析対象事例に依存しない新たな設計領域知識となる。

4.3 説明に基づく学習の導入

4.3.1 説明に基づく一般化 (EBG) の問題点の解消

2.4.1節に述べたように, EBG を応用した知識獲得システムを構築するときには, 「知識獲得のための知識獲得ボトルネック」ならびに獲得された知識は「真の意味での新しい知識」ではないという問題が生じる。本研究では, これらの問題点に対して, つぎの二つの方法で対処する。

(1) 一般的な領域知識の記述 領域知識に対して, 予め全て完全に用意しておくのではなく, 実在事例の説明に用いて, 自らが精錬されていくような性質を与えることによって, EBG 手法の問題点を解決することができるものと考える。したがって, システムの初期状態において用意されている領域知識は, 説明対象事例に含まれる情報が加えられて初めて現実的・操作的になるように, 一般性をもたせておく必要がある。たとえば, つぎの知識を考える。

> 測定対象 x の物理量 (またはその変化) p を測定器の構造要素 y の物理量 (またはその変化) q で取り込む (detect) ためには, 少なくとも x, p, y, q の間になんらかの物理法則 (p_law) が成立していなくてはならない

$$detect(x,p,y,q) \leftarrow p\_law(x,p,y,q)$$

この表現を含意則に読み換えると, 任意の物理法則が成立すれば測定器中の detector の機能を果たすことができると解釈することができる。ここで説明対象事例を用いて演繹推論をガイドする必要性が生じてくる。この推論により, 測定装置の detector としての機能がどのような物理法則の成立によって支えられているかについての知識が獲得される。この知識は, ガイドした事例と同じメカニズムを用いて測定するような設計物にしか応用できない「狭い」知識になってしまう反面, もしそのような設計物を事例として与えた場合, その説明木生成を格段に早くする機能をもつ。また, これによって得られた知識は, もはや説明対象事例によって非現実的にならないようガイドされる必要のない知識であり, 設計解候補の機械的な生成を可能にする領域知識であると考えることができる。

このように, 一般的知識を実際に存在する設計物の説明に使用することによって, その設計物 (事例) にガイドされながらこれら知識が具体化され, 実用レベルでの利用性の高い知識が生成される。ここで生成される知識は, 単に領域知識を組み合わせただけの知識ではなく, 説明対象事例に含まれる情報を使って初めて生成される知識である。すなわち, EBG 手法では単に既有の知識を組み合わせただけの知識しか生成し得ないとする問題に対するひとつの解答が与えられる。


```
ebg(A,_,_,_,_)            :- call(A).
ebg(A,Ga,[ex(A)],[Ga],_) :- ex(A).
ebg(A,Ga,[A],[Ga],_)      :- dt(A).
ebg((A,B),(Ga,Gb),P,Gp,Tr) :-
    ins([x,B],Tr,Tre), ebg(A,Ga,Ap,Gap,Tre), append(Ap,[x],W),
    ins(W,Tr,Tree),ebg(B,Gb,Bp,Gbp,Tree), append(Ap,Bp,P),append(Gap,Gbp,Gp).
ebg(A,Ga,[P],[Gp],Tr) :-
    dt(Ga,Gb), copy(dt(Ga,Gb),dt(A,B)), ins([[A,x]],Tr,Tre),
    ebg(B,Gb,Bp,Gbp,Tre), append([A],Bp,P), append([Ga],Gbp,Gp).
ebg(A,Ga,[A],[Ga],_)      :- !,fail.
copy(Old,New) :- !,assert('$marker'(Old)),retract('$marker'(New)).
```

図 4.2: EBG エンジン (一部)

本研究では, さらに 4.5節で, より一般性の高い知識表現についても考察する。

(2) 説明木の部分的チャンク化 一つの説明木から生成される知識が, 予め特定された目標概念と, 予め設定されている操作性規範との間をつなぐバイアス的知識だけであることは効率的ではない。また, 2.4.1節で述べたように, 操作性規範は設計局面や設計者の意図によって動的に変化するものであり, 操作的であることの基準が予め固定されていること自体にも問題がある。このような理由から, 本システムでは, 説明木の任意のノードに目標概念を, 任意の空間に操作性規範を設定することができる機能を加えた。これによって説明木の任意の部分木に注目し, 1 つの説明木から複数の知識を抽出して, 領域知識に加えることができる。

この説明木の一部から抽出された知識は, 数段の木構造を一まとめにして (チャンク化) 一段の知識とするものである。説明木からの知識抽出方法については, 4.4節で述べる。

4.3.2 論理プログラミングによる EBG の実装

本研究では, 2.4.1節で述べた EBG システムを参考にして prolog のメタインタプリタにより説明木生成エンジンを実現した[1]。

図 4.2に, 作成した説明木生成および一般化エンジンを示す。なお, 領域知識の記述方法は通常の節形式 (`a :- b,c,d.` または `e.`) から `dt(a,[b,c,d]). dt(e).` のように変更した。同図の上から三行目までは, 説明木の「葉」を認識する節である。第 1 引数と第 2 引数は, 変数を含んだままの, 現在探索中のノードである。それが prolog の組み込み述語なら第 1 行目にマッチする。説明対象事例を記述するゴール節 (入力の構造・属性記

[1] 以下, prolog の書式に習って, 大文字で始まる文字列を変数とする

述) と単一化するなら第 2 行目にマッチする。領域知識に含まれるゴール節 (一般的な事実 fact) と単一化するなら第 3 行目にマッチする。いずれの場合も, ゴール節と単一化されるのは入力事例に依存した第 1 引数 A だけであり, 第 2 引数 Ga は, 変数を残したままである。第 3 引数と第 4 引数は, 生成した説明木の葉の部分をこの節を呼びだした上位の節に渡すための変数である。ここで, 第 3 引数には A を, 第 4 引数には Ga を置くことで, 第 3 引数は変数を含まない具体的な説明木の「葉」, 第 4 引数は変数を残したままの説明木, すなわち一般化された説明木の「葉」とすることができる。

図 4.2の第 4 節は, 複数の条件の連言をつぎのレベルで一つずつ説明して行くためのものである。第 5 節は, 説明構造のもう一段先まで推論を進めるための節である。現在探索中のノードと単一化する事実 (ゴール節) が領域知識中の一般的な事実にも, 入力事例の記述にも無く, 操作性も満たさない場合, すなわち説明木の「葉」になり得ない場合, さらに別のルールを用いて一般的な事実または入力として与えられた事実による説明に到達しなければならない。そこは説明木の分枝点となる。先ず, そのノードを成立させる十分条件を探索する (dt(Ga,Gb))。Ga,Gb は変数を含む項である。これを, 述語 copy により複製し, 変数が単一化される項 A, B を生成する。

述語 copy により Ga, Gb に含まれる変数は変数のままその構造だけが A, B と単一化される。これにより, 節に含まれる変数は入力事例に依存した特定の定数に単一化されることなく, 同一物体であるか否か, 同一変数であるか否かという構造的な情報のみが保存される。ここで説明木の分岐は親となるノードの直後にあるリストによって表現される。リストのそれぞれの要素は, 再帰的に呼び出される一つ下のレベルの節で生成された部分的な説明木によって満たされる。このリストは, 中の構造としても同様の構造をもつので入れ子になったリストは内側のものほど多くの分岐回数を経たものになっている。

図 4.2の第 6 節は, 枝の探索が失敗した場合に無限ループに陥らないようにするために導入されたものである。すべての節が第 1 行目から第 3 行目までにマッチすれば, 目標概念に対する説明木が生成され, 同時に一般化された説明木が知識として獲得される。

本研究では, 説明木生成エンジンに以下に示す機能を加えた。すなわち, 説明木生成過程をユーザに提供する視覚化プロセス *visu*, 全体の説明木完成を待たずに説明木の一部分をチャンク化することのできる学習プロセス *learn*, 「根」の方から現在探索している枝までの説明木を保存するプロセス *ins* を導入した。図 4.3に主なサブルーチン・プログラムのリストを示す。

説明木生成プロセスの視覚化 4.3.1節に述べたように, 高度な一般性をもった設計知識を予め用意しておき, 事例の説明に用いて事例に含まれる情報を加え, 現実的, 操作的な方向に知識を精錬するアプローチによって「真の意味で新たな知識が獲得されるか」と


```
ins(A,[B,x],[B|A]) :- !.
ins(A,[x|B],[A|B]) :- !.
ins(A,[B|X],[C|X]) :- ins(A,B,C).
ins(A,[X|B],[X|C]) :- ins(A,B,C).

mkrule(H,L,(H :- B)) :- conjunct(L,B).
conjunct([X],X).
conjunct([X|Y],(X,Z)) :- conjunct(Y,Z).

append([],Y,Y) :- !.
append([A|X],Y,[A|Z]) :- append(X,Y,Z).
```

```
visu(Tr) :- v(Tr,1),get0(_),!.

v([H|T],N) :- N1 is N+4 ,
              v(H,N),dev(T,N1).
v((H,T),N) :- v(H,N),v(T,N).
v(ex(H),N) :- tab(N),write(H),nl.
v(H,N)     :- tab(N),write(H),nl.

dev([],_)    :- !.
dev([H|L],N) :- v([H],N),dev(L,N).
```

図 4.3: 主なサブルーチン

いう問題に対する解答を得ることができる。しかし, その反面, この知識だけを使って説明木を完全自動で生成することは困難になる。したがって, 領域知識がある程度まで拡張されるまでは, 説明木生成の制御をユーザに委ねざるを得ない。そこで, 本システムの説明木生成モジュールに, 説明木生成過程を視覚化し, ユーザに提示することによって, 説明木生成を遅らせている原因となっている知識に対する補修, または知識の追加等を可能にする機能をつけ加えた。

図 4.3に示す述語 *ins* は, 木構造 A を木構造 B の中にあるマーク x の部分に挿入した結果を第 3 引数と単一化する。述語 *visu* は, リスト表現された計算機内表現を木構造に整形して表示する。この 2 つの述語によって, 説明生成過程を視覚化することができる。

説明木生成プロセスの途中で領域知識を抽出する機能 EBG 手法は, 元来, 説明木生成→一般化→操作性規範適用→知識抽出という段階を踏むものである。しかし, 同一事例の説明木の中に同じ説明構造が複数存在する場合には, 説明木完成を待たず, 説明木生成途中における部分木からの知識抽出が有効と考えられる。本システムの説明木生成モジュールは, 説明木生成途中に, 部分的に完成した説明構造から述語 (*mkrule*) によって知識を抽出する機能をもつ。述語 *mkrule* は, 説明木の葉の部分の述語を集めたリスト構造を連言の形に変換してこれを条件部 (ボディ部) とし, 目標概念に相当する述語をヘッド部にしてホーン節を生成するもので, これによってチャンク化されたマクロルールが生成される。

4.3.3 物理的設計物に対する領域知識と説明対象事例

領域知識に含まれる規則 (rule) は含意則の形式をもち, prolog のホーン節によって記述することができる。また, 領域知識に含まれる事実 (fact), および説明対象事例は, prolog のゴール節によって記述することができる。本節では, 3.3節で述べた物理的設計物の成り立ちを説明するための階層型設計知識空間 (G, P, F, L, S) のそれぞれを, EBG

の概念 (ターミノロジー) と対応づけながら説明する。

目標集合空間 (G) 測定機器の目標は, 測定対象のもつ物理量を測定することであり, 測定とは, 「ある量を, 基準として用いる量と比較し, 数値または符号を用いて表す」ことである [63]。換言すれば, 「測定する」ことは, 「ある対象の物理量を信号または (外部のシステムに認識可能な) 測定器内部の物理量として出力する」と解釈することができる。たとえば, 「質量を測定する (*measure_mass*)」という要求は, 以下のように記述される。

```
measure_mass ← measure(Object,mass,X,shift).
```

与えられた物体 (*Object*) の質量 (*mass*) を何か (*X*) の変位 (*shift*) として測定 (*measure*) する

可撓棒を例とした場合,

```
measure_fluid_velocity ←
      measure(Liq,velocity,X,change_in_resistance),phase(Liq,liquid).
```

流速を測定する (measure_fluid_velocity) ための必要十分条件は, 与えられた流体 (Liq) の速度 (velocity) を計測 (measure) することである

が目標集合空間 (G) から選ばれる。これは, EBG 法の目標概念に対応づけられる。

さらに, 付随的な目標, たとえば耐環境性, 測定範囲の局所性なども, この空間上の知識である。可撓棒の例では, 目標は基本機能「流速の測定」であるが, 同時に「溶融鉛の高温に耐えること」, 「測定範囲を局所化すること」も満足しなければならない。

プラン型知識空間 (P) ある機能は副機能列に詳細化することができ, その機能が達成されるか否かは展開された副機能列の達成によって決定されると考えられる。物理因果関係に展開されるまで機能から副機能への詳細化がおこなわれる。すなわち, 物理因果関係は機能達成を説明するプリミティブであり, 物理因果関係の連鎖によって機能達成が根拠付けられる。このような根拠付けの際に, 無数に存在する物理因果法則の組合せ (連鎖) を盲目的に調べることは非現実的であり, 基本機能達成のために展開すべき副機能をトップダウンに規定するために必要な知識が, いわゆるプランであると考えられる。

一般に, 測定機器領域には様々なプランが存在するが, 図 4.4 に示す偏位法と零位法が代表的である [63]。偏位法では, 「測定対象の測定したい物理量を, 測定器の一部の物理量として検出し, 変換器まで伝送し, 指示器によって, 所定の情報に変換する」という機能の流れで測定という目標を達成する。零位法では, 「測定対象の測定したい物理量を, 測定器の一部の物理量として検出し, 変換器まで伝送すると同時に, 既知量を変換器まで伝送し, それぞれの物理量を平衡器によって平衡を確認し, 既知量から測定量を知る」

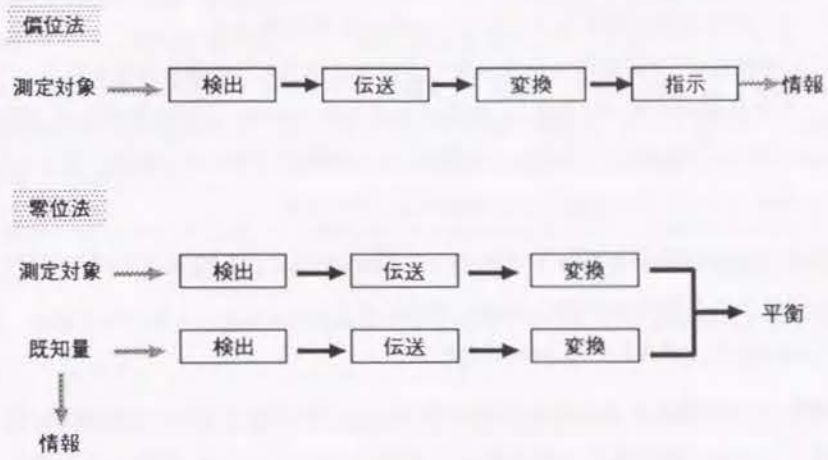


図 4.4: 代表的な測定プラン

という機能の流れで測定という目標を達成する。他にも, 零位法の一種として知られる自動平衡法では, 比較平衡と既知量の調節を自動的に行う。このように, 測定器に要求される機能も測定方式により異なる。偏位法を例にとると, そのプランは図 4.4に示すように「検出→伝送→変換→指示」の機能列となり, 以下のように記述される。

```
measure(X,P,T,R) ← detect(X,P,Y,Q),transmit(Y,Q,Z,R),transduce(Z,R).
```

任意の測定対象 (*X*) の物理量またはその変化 (*P*) が測定されるための十分条件は, 対象 (*X*) の物理量またはその変化 (*P*) を, 設計物の構造要素の 1 つ (*Y*) の物理量またはその変化 (*Q*) として検出し (detect), それを構造 (*Z*) の物理量またはその変化 (*R*) に伝え (transmit), そこで外部 (人間または他のシステム) に利用可能な情報 (*R*) に変換すること (transduce) である。

可撓棒の例では, 変数 P, Q, R に, それぞれ流速 (velocity), 力 (force), 歪 (strain) が単一化されている。

このように高度に一般化されたルールを含むシステムによって導出されるプランあるいは機能群の数は, 一般に膨大となり, その中には実際の環境下では観測されないような非現実的なものも含まれる。

本節で提案する知識獲得システムは, この一般的抽象的な知識を現実的操作的なレベルの知識に特殊化する。この特殊化は, 事例に依存した方向に成されたものであり, それゆえに, この特殊化された知識の中に, 設計に盛り込まれたアイデアを見ることができる。

機能知識空間 (F) 一般に測定機が必要とする機能には, ある物理量を信号として検出すること (detect), 信号の値を他の値と比較すること (compare), 信号の値と他の値との平

衡を取ること (balance), 信号を別の場所へ伝えること (transmit), 信号を変換しつぎのモジュールで利用可能な形にすること (transduce), 等がある。

これらの機能は, 下位機能や物理現象の連鎖およびそれらの組み合わせによって達成される。この空間内の各々の知識は, 機能が達成されるために必要な物理法則, 機能が達成されるための上位機能と下位機能との関係, および機能そのものの表現に関するものである。したがって, ここで定義される機能とは, たとえば

> 物体 X の物理量 P を物体 Y に伝達する (transmit) には, 物体 X から Y への物理量 P の位置的な伝達 (l_trans) を行えばよい
> `transmit(X,P,Y,P) ← l_trans(X,P,Y,P).`
>
> 物体 X の物理量 P を物体 Z の物理量 Q として伝達する (transmit) には, 物体 X において物理量 P を物理量 Q に変換 (p_trans) した後, 物体 Y に伝達すればよい
> `transmit(X,P,Y,Q) ← p_trans(X,P,X,Q), transmit(X,Q,Y,Q).`
>
> 物体 X において物理量 P を物理量 Q に変換するには, 物理量 P と物理量 Q との間に, 物体 X に依存した物理法則 (p_p) があればよい
> `p_trans(X,P,X,Q) ← p_p(X,P,X,Q).`

のようなものである。ここに示した高度に一般的な定義をもつ機能集合に対して, 設計物に含まれている物理法則または下位機能の中から, 与えられた目標を達成するために必要な手段が探索される。可撓棒の例では P, Q 等の物理量に力 (force) や歪 (strain) 等が単一化されて説明木中に現れる。たとえば, detecter としての機能が達成されるための十分条件はつぎのように書くことができる。

> 構造 X の物理量 (またはその変化) P が構造 Y の物理量 (またはその変化) Q で取り込まれるための一つの十分条件は, X, P, Y, Q 間にある物理法則または効果が成立していることである
> `detect(X,P,Y,Q) ← p_law(X,P,Y,Q).`

しかし, 如何なる物理法則が成立していようと, それが必ずしも detecter としての機能を構造物に提供するものではない。言い換えると, detecter としての機能の達成を裏付けることができる物理法則は自ずと限定されている。しかし 3.3.4節で示したように, 物理因果法則空間は現存する物理法則・数理法則を網羅した上に, それらを任意に組み合わせることができると想定している。このような無数の物理因果法則の各々に対して, その法則が detecter としての機能を裏付けることができるかどうかをを予め与えておくことは事実上不可能である。ここに, 実在する設計物を説明対象事例として EBG 法を適用し, 事例に依存した知識を獲得しながら領域知識を精錬する理由の一つがある。

物理因果法則空間 (L) 機能集合空間上のそれぞれの機能は物理現象の因果連鎖によって成立している。物理法則空間では, 一般に物理法則がどのような条件下で存在し, どのような物理量の間に存在するかに関する知識を, 一般則 1, 2 (3.3.4節, 図 3.4 参照) の一部を特殊化した形で表現する。図 3.4に示した 2 つの一般則は, ホーン節としてつぎのように表すことができる。

```
physical_law(X,P,X,Q) ← has(X,P),p_property(X,P,Q),assert(has(X,Q)).
physical_law(X,P,Y,Q) ← has(X,P),p_relation(X,P,Y,Q),assert(has(Y,Q)).
```

これを, 本研究で作成した説明木生成エンジンに使用することができる形式に変形するとつぎのようになる。

```
dt(p_law(X,P,Y,Q),[has(X,P),p_p(X,P,Y,Q),assert(has(Y,Q))]).
```

ここで物理特性 (p_property) と物理関係 (p_relation) の違いは, 物理特性に関する構造物が X のみであるのに対し, 物理関係に関する構造物が, X と Y の 2 つであることである。これら 2 つの一般型は, prolog の単一化機構を利用して, 1 つの統一的な述語 p_p(physical_property & physical_relation) で表現している。

この表現で, has(X,P) と assert(has(Y,Q)) は, 2 つの一般則に共通のものであり, かつ, 暗黙裏に成立を仮定することができる。そのため, 実際に機能集合空間と構造属性集合空間を結び付ける役割を果たしているのは, p_p(X,P,Y,Q) のみであると考えることができる。たとえば, 「可撓棒」の流速測定の説明に使用された p_p の一部をつぎに示す。

> 弾性性質 (elastic) をもてば, 応力 (stress) と歪 (strain) に因果関係が成立する
> `p_p(X,stress,X,strain)← hasp(X,elastic).`
>
> 流体 (liquid) の中に固体 (solid) が挿入 (locate_in) されておれば, 流速 (velocity) と固体に働く力 (force) には因果関係が成立する
>
> ```
> p_p(L,velocity,S,force)←
>     locate_in(S,L),has(L,velocity),phase(L,liquid), phase(S,solid).
> ```

異なる 2 つの物理因果関係 p_p に含まれる変数に, 同一の実体を単一化することによって, 物理因果連鎖が構成される [64]。図 3.5に示した可撓棒の測定プロセスを説明する物理因果連鎖では, L1 の固体 (Y1) と L2 の弾性体 (Y1) が, 共に事例中の obj4a に特殊化されることにより L1 と L2 が連結されている。個々の因果法則 (L1～L6) は, 機能系統図の末端機能 (図 3.2 F1～F6) の達成を根拠付ける。

構造知識空間 (S) この空間には, 測定器の構造や属性に関する知識が記述されている。すなわち, 構造・属性に関する一般的な事実 (無条件に成立する関係など) や, *is_a*, *part_of* 関係による性質継承等の知識がこの空間に存在する。領域知識のゴール節が事実を表す。

たとえば,「ガラスは固体である」phase(glass,solid).「石英ガラスはガラス (の一種) である」is_a(quartz_glass,glass). 等である。また「X の相 (phase) P は X の材質 (substance) に依る」phase(X,P) ← substance(X,S), phase(S,P). という一般的な知識を用いて,「X の材質 (substance) がガラス (glass) で, ガラスは固体 (solid) であるから, X は固体である」といった構造的な推論がおこなわれる。

説明対象事例

本研究では, 既存設計物の表層的知識から, 内包されている設計知識を抽出することを目的としている。そのため, システムの入力情報となる設計事例の表現は, その表層的構造特徴に限定し, これを 図 3.10(b) で定義したネットワーク形式で記述する。すでに図 3.1に,「可撓棒」の構造的特徴を意味ネットワーク形式でコーディングした結果を示した。この形式で記述された対象物は, アーク (arc) のラベルを述語名, そのアークで結び付けられるノード (node) を引数とした統一的な形式でシステムに登録される。

4.3.4 機能系統図の一般化

以上のような領域知識を使って, 説明木, すなわち機能系統図が生成される。図 4.5に示すのは, 試作した知識獲得システムが生成した, 可撓棒に対する機能系統図である。整理された目標集合, 機能集合, 構造属性集合や, プラン, 物理法則に関する知識のみを用いて説明がなされているため, 用語が統一された表現となっている。

システムは, この説明木から, 訓練事例に依存した引数を選択的に変数化し (identity elimination), 図 4.6に示す一般機能系統図 (GFD) を生成する。たとえば, 構造要素 o4 の相が固体であるという記述 (phase(o4,solid)) のうち, 固体 (solid) という属性は一般化してはならないが, 構造要素 o4 は同じ属性をもち, 他の要素との関係も同じであれば必ずしも o4 でなくてもよいので変数化される。同様に, 物理法則が弾性性質という属性を要求している構造要素 o4a は, 弾性体である限りその材料はどのようなものでもよいので, o4a の材料 quartz_glass も変数化される。

この一般的な説明木には測定に必要な情報のすべてが含まれ, 余分な情報は EBG のもつ目的指向型フィルタリングの機能により除去されている (irrelevant feature elimination)。その結果, 事例では流体は鉛であったが, 測定機能の説明木に物質が鉛であるという属性は使用されず流体であるという属性だけが使用されている。したがって, この測定器からは鉛に限定されない一般の流体の測に対する説明木が生成されることになる。

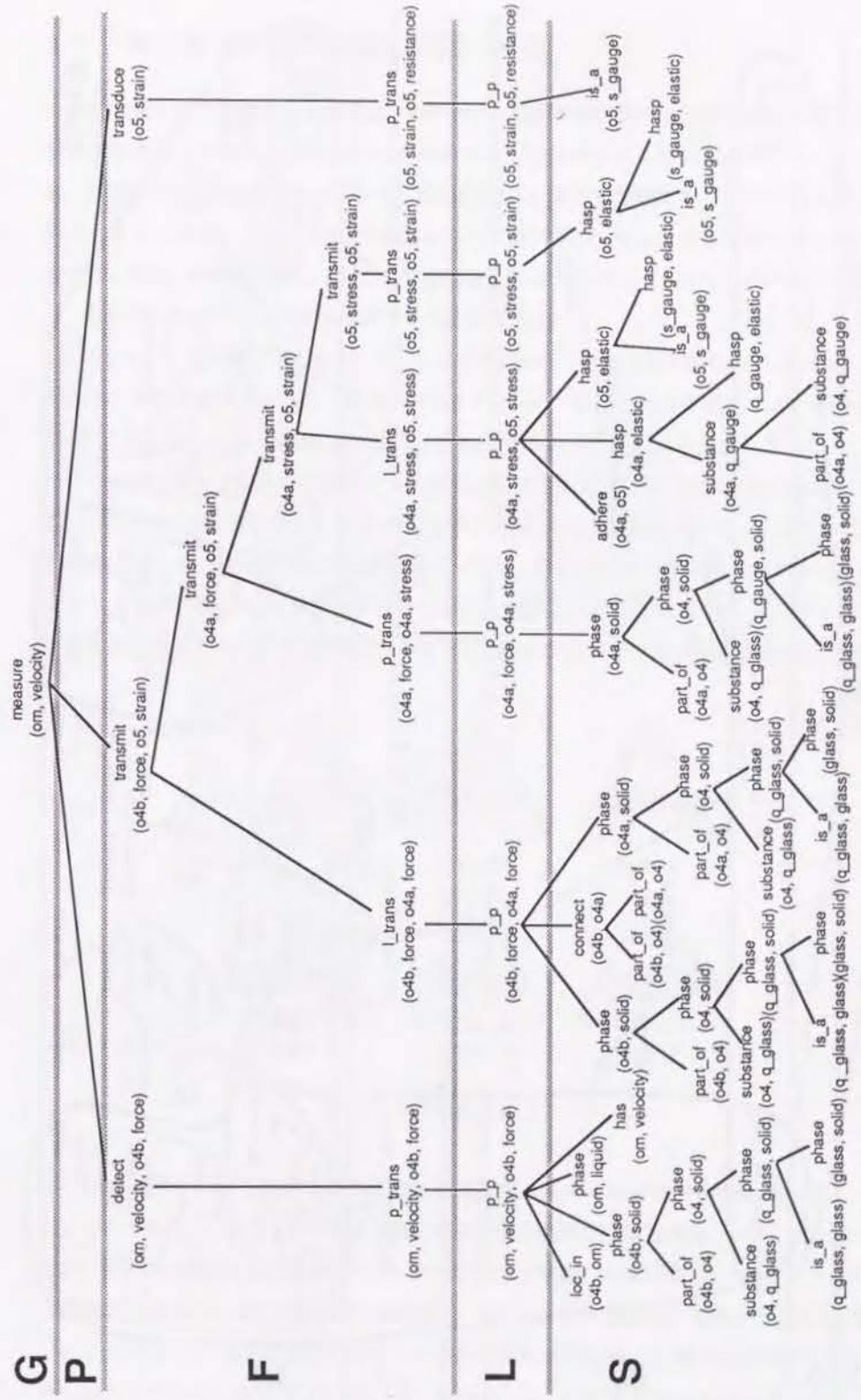


図 4.5: EBG によって生成された「可撓棒」の説明木

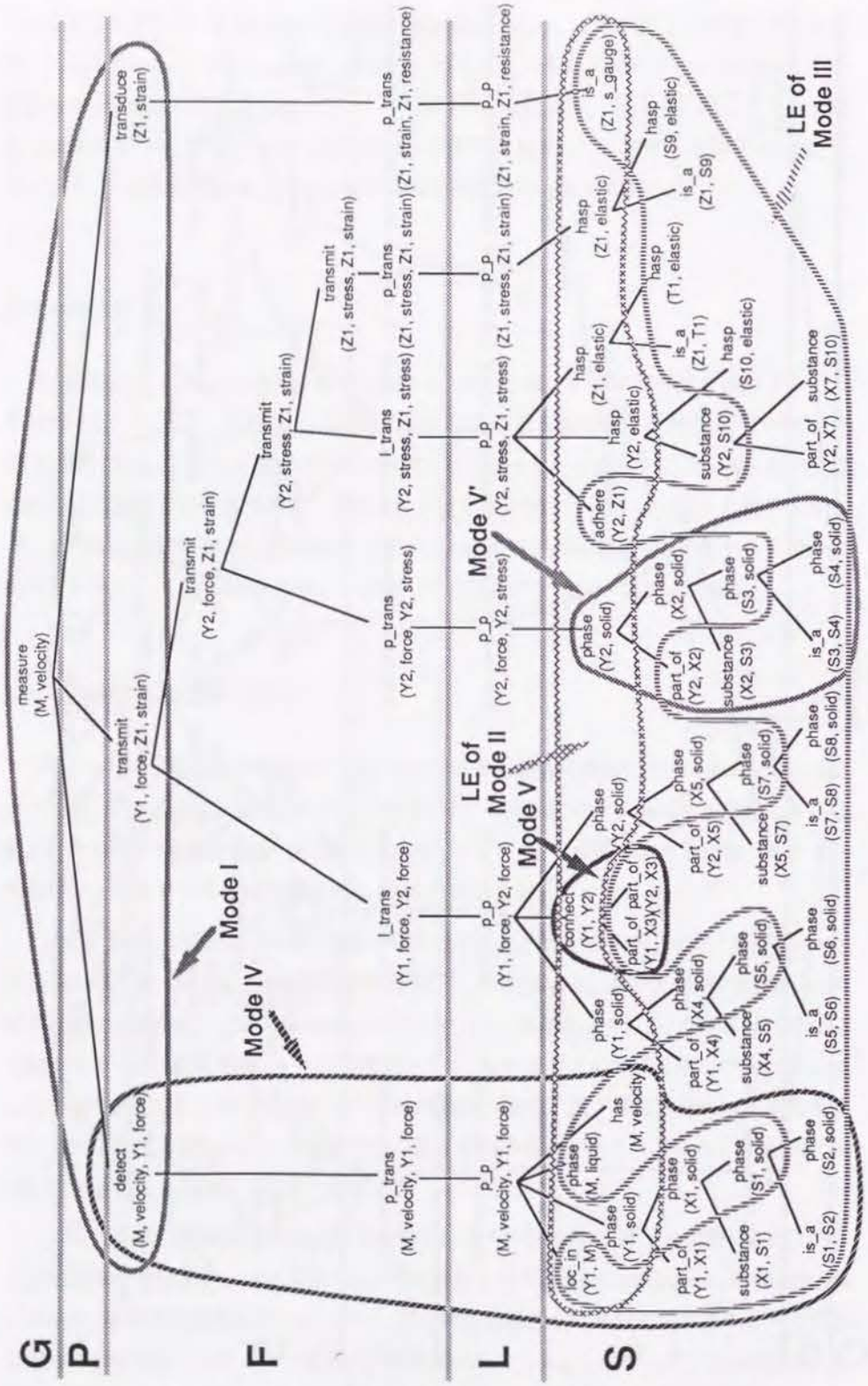


図 4.6: 一般化された「可撓棒」の説明木と 5 種の知識抽出モード

4.4 単一事例からの多様な知識の獲得

Mitchell らの提案した手法では, 一般化された説明木をチャンク化 (途中の説明を省略し葉の部分だけを取り出す) して, 目標概念の十分条件とする知識を獲得する。すなわち, 目標概念と (予め与えられている) 操作性規範を満たす空間間のバイアス的知識を獲得するものである。これに対し, 学習対象概念を説明木の「根」にある目標概念のみに固定せず, また, 操作性規範についても適宜変更することによって, 説明木の一部分を抽出し, そこから獲得された知識も有用であると考えられる。

すなわち, 2.4.1節に示したように, 設計者が知識をどのような設計局面に適用し, どの段階まで設計を進展させることを意図するかによって, 操作性規範は変化する。本研究において, 設計領域知識を機能, 物理現象, 構造・属性の三空間およびそれらをつなぐ二つの媒介知識空間から成る多階層モデルに整理して表現したことは, それらから生成される GFD に設計知識空間の違いによる境界 (操作性境界 [65])) を発生させる。したがって, 目標概念 (GC) を一般機能系統図の任意のノードに, それを説明づける表現レベル (以下説明レベル (LE) と呼ぶ) を操作性境界のいずれかに設定することによって, 図 4.7 に示す I ～V の獲得形式 (モード) に代表されるような, 様々な形式での知識獲得が可能になる。

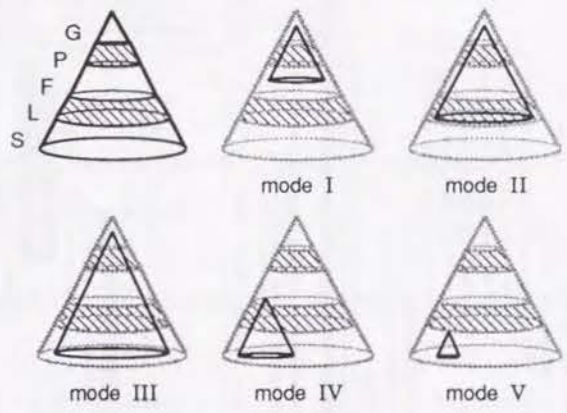


図 4.7: 5 つの知識抽出モード

図 4.8に示すのが, 実験用に試作した知識獲得システムのグラフィックユーザインタフェースである。左上のウィンドウが解析対象事例の表層的な構造情報 (一部), 右上のウィンドウが領域知識 (一部), 左下のウィンドウが機能系統図の生成過程, 右下のウィンドウが一般機能系統図の生成過程を示す。操作者は, 表示された木の任意の部分木を選択し (ウィンドウ上では, GC と LE に属するノードが白黒反転表示される), 新たな知識として抽出することができる。図 4.8の右下のウィンドウは, GC を目標 G ($measure(X, velocity)$) に, LE を構造属性空間の最下層に設定してチャンクしている状態を表している。

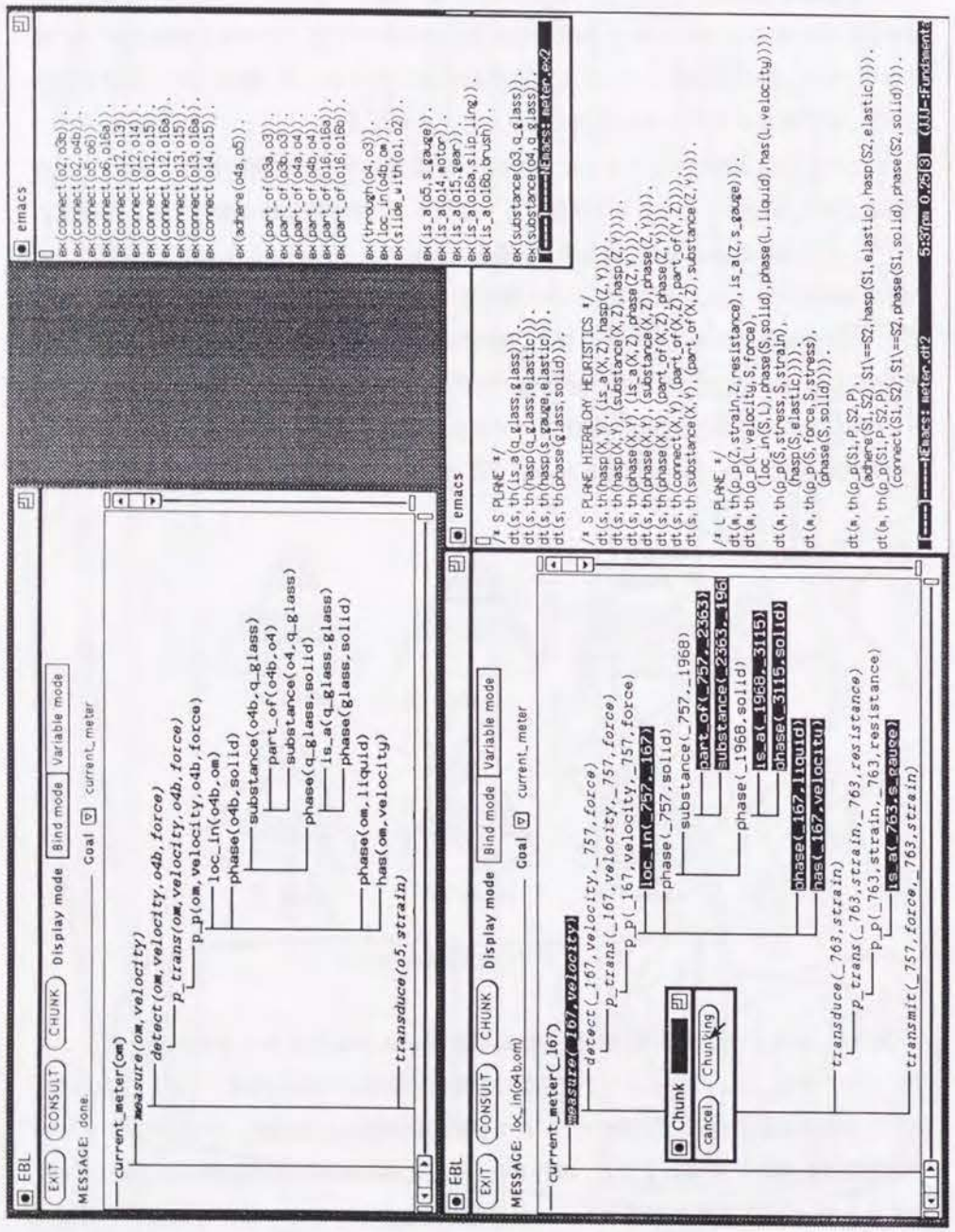


図 4.8: 知識獲得システムのグラフィックユーザインタフェース

モード I (プラン型知識の抽出)

モード I は, GC を目標集合空間 (G) に, LE を機能知識空間 (F) の最上位レベルに設定する。この場合, 設計物の基本機能がどのような部分機能から成り立っているかについての知識を得ることができる。つまり基本機能を実現するための基本的な機能展開プラン についての知識を得ることができる。

可撓棒の場合, 図 4.6に網かけ線で囲った部分 (mode I) から知識を抽出する。M が Liquid(液体) を表しているものと考えると, 図 4.9(a) に示す知識 (流速測定の十分条件), すなわち, 「流速を測定するという目標を, 流速を可撓棒の先端部で棒の受ける力として

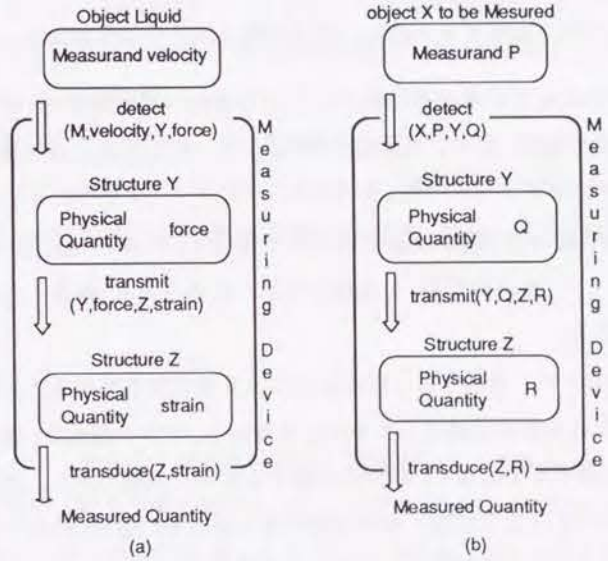


図 4.9: 獲得された流速測定のためのプラン型知識 (a) と一定的な測定プラン (b)

検出し, これを歪いう形で歪ゲージまで伝達し, 歪ゲージで電気信号に変換するという機能に展開する知識」を獲得する。

この知識は, 予めプラン型知識空間にあった一般的なプラン (図 4.9(b)) を事例に従って具体化したものになっている。すなわち, EBG で得られる知識は事例を一般化したものであると同時に領域知識を特殊化したものと見ることができる。獲得された知識は予め与えられていた知識と較べてより制約されたものであり, 適用することができる範囲は狭くなっている反面, 実在する事例にガイドされて導出された知識であるため, その正当性は保証されたものであると考えることができる。

一般に, 目標概念と知識抽出レベルとの間のプラン型知識空間 (P) や物理法則知識空間 (L) が, それぞれ目標合理性や因果合理性を満足するものであれば, 獲得された知識も

これらの合理性を有する。しかし, 空間 P に因果合理性を有する領域知識をあらかじめ用意しておくことは非常に困難である。なぜなら, 利用される物理法則の組み合わせが莫大な数になり, 各々の物理法則について多数のケースを試みてみなければ因果合理性は保証されず, これは事実上不可能だからである。そこで本研究では, 空間 P の領域知識は目標合理性のみ保証されるような一般性の高い知識とし, 説明生成プロセスに基づいて空間 L あるいは S の知識と共にコンパイルされて初めてに因果合理性も保証される, というアプローチをとった。つまり, 事例となる設計物を与えることによりそこに含まれる合理性を用いて (因果的には) 過度に一般的なプラン型知識から実用的な設計プランを得ることができるのである。

モード II (目標機能を達成するための一般的な構造的十分条件の抽出)

モード II は, GC を目標集合空間 (G) に, LE を構造・属性空間 (S) の最上位に設定するものである。この場合, 測定に最低限必要な, 最も一般的な測定器の構造・属性が得られる。

可撓棒の場合, 図 4.6中網かけ線で囲まれた部分 (LE of mode II) に含まれる述語 (atom) の連言として, 流速測定の (可撓棒の例から含意される) 構造上の最も一般的な十分条件が獲得される。

この知識は, モード I のように目標合理性のみを満足するレベルにおいて説明木生成を中断したものとは本質的に異なっている。すなわち, プラン型知識, 機能集合空間中の知識, 物理因果法則のみではなく, 事例の知識をも含んだ知識である。また, この知識は, 以下に示すモード III (LE を構造・属性空間の最下層とした場合) において得られる知識を抽象化し, コンパイルした知識とみなすことができる。

モード III (目標機能を達成するための具体的な構造的十分条件の抽出)

モード III は, GC を目標集合空間 (G) に, LE を構造・属性空間 (S) の最下位に設定するものである。II が最も一般的な (適用範囲の広い) 知識表現を指向するのに対して, この場合, 入力事例に細部の構造まで依存した, 最も具体的な知識が獲得される。本例題では流速測定という目標を達成するためにどのような構造・属性を持てばよいかについての具体的な記述が得られる。これは, 図 4.6中網かけ線で囲まれた部分 (LE of mode III) に含まれる述語の連言として与えられる。

この記述は, 入力された元の意味ネットワーク表現と比較すると, 目標達成に無関係なノードをフィルタリングにより落とし, 関連のある領域知識中の記述を付加した, 基本機能達成のために必要最小限の意味ネットワークとなっている。

モード IV(機能-構造間の連関知識の抽出)

モード IV は, GC を機能空間 (F) に, LE を構造属性空間 (S) に設定する。この場合, 概念設計に不可欠な機能達成のための構造的条件知識が獲得される。

可撓棒の場合, たとえば目標概念を detect(M,velocity,Y1,force), 説明レベルを構造属性空間の最上位に設定すると図 4.6の網かけ線で囲まれた部分 (mode IV) が抽出される。ここから「速度をもつ物質の相が液体である場合, 固体の一部をその中に浸せば, 速度をその固体に働く力として検出することができる」という知識を得ることができる。

もともと用意されていた知識「何等かの物理法則が成立していれば detector としての機能を達成する」は, 制約が弱すぎてそのままでは現実の設計に適用できない。これが事例にガイドされることによって, 片持ち梁の自由端を流体中に挿入したときに成立する物理法則に基づいて, この構造が detector としての機能を果たすことができるという, 一つの設計知識を得ることができる。この物理法則が成立するためには, 少なくともどのような構造を形成していればよいかについての条件が, 獲得された知識の条件部となっている。このように, "検出" のための "機能-物理法則-構造・属性" の連関知識が獲得される。

モード V(構造モジュールの抽出)

モード V は GC, LE を共に構造属性空間に設定するものである。この場合, 1 つの構造モジュールとも呼ぶべき知識が獲得される。たとえば, 図 4.6において網かけ線で囲まれた部分 (mode V) からは, 「Y1 が X3 の一部で, Y2 も X3 の一部であれば, Y1 と Y2 は物理的に結合している」という知識が得られる。

この形式の知識は比較的抽象度の高い構造をより具体的な構造として決定する場合に適用される。とくに, 4.3.2節で提案した説明木生成途中での知識抽出が有効なのは, このモードで獲得された知識である。

たとえば, 図 4.6において網かけ線で囲んだ説明木の一部分 (mode V') に注目する。これは, ある物体 Y2 が固体であることの説明木である。三段の木構造をもつこの説明を生成するためには, 何回かの fail と back tracking を繰り返すことになる。過去に説明した事例の数が増えるにしたがってそこから獲得された領域知識の数が増大して行き, それに連れて説明木生成に費やされる記憶容量および時間も増大する。そこで一旦説明に成功すれば, そこでチャンク化しておき, 同一事例中に同じ説明構造を必要としている部分があればその知識を直接適用することができるようにする。これによって説明木生成の労力を削減することができる。

獲得した知識の利用

ここに示した様々な知識抽出形式は, 種々の設計局面における設計者の要求にしたがって適切なものが選択, 利用される。また, I と IV, II と V の形式のように組み合わせて使用することもできる。たとえば, ある設計物から得た I の知識と別の設計物から抽出した IV の知識を組み合わせて使用する場合には, プランを機能に展開する方法のみを一つの設計物と同じ方法で行うが, 下位機能を達成する構造としてはその設計物とは別の実現方法を用いることができる。このようにして, 因果合理性の各段階に別々の設計物から抽出したものが使われて得られる設計物は, 学習事例のどれにも似ていない新しい設計物であるといえる。

このような獲得された知識の利用については, 6 章で述べる。

4.5 決定則の特殊化による設計知識獲得

本章前半では, 領域知識を演繹的に組み合わせた記述が真に新たに獲得された知識といえるか, という EBG の問題点を解決するために, 初期領域知識として一般的な設計知識を与えておき, それに事例記述の情報を加えることで, 領域知識ベースを洗練するアプローチを提案した。本節では, さらに一般的な設計知識表現形式として決定関係 [11] に注目し, 設計領域における利用方法について理論的側面から検討する [66][67]。EBG との関連では, 決定関係は, 事例の説明木をソースとして領域知識を生成する背景知識とみることができる。

4.5.1 EBG における領域知識の役割

EBG によって獲得された知識は, 領域知識を演繹的に組み合わせ特殊化した結果という側面と, 単一事例の記述を一般化 (SIG: Single Instance Generalization) した結果という側面をもつ。前者に関しては, 演繹推論によって導くことができる知識をわざわざ知識ベースに加えることの冗長性から, 知識探索効率の低下が問題となる。したがって, 多くの演繹可能な知識の中から, 探索効率を低下させてまで生成しておくべき価値のある知識を指定するためのガイドが必要となる。後者に対しては, 一般には論理的な正当性 (妥当性) が保証されないことが問題となる。

SIG によって生成された知識に正当性を与える一つの方法として, 健全な推論を行なわせるような背景知識を与えることが考えられる。EBG 手法はこれらの問題を相補的に解決しているという見方ができる。すなわち, 図 4.10に示すように, 演繹推論による知識生成は実在する事例の記述にガイドされ, SIG は領域知識によって正当化されている。この枠組においては, 領域知識のもつ性質として

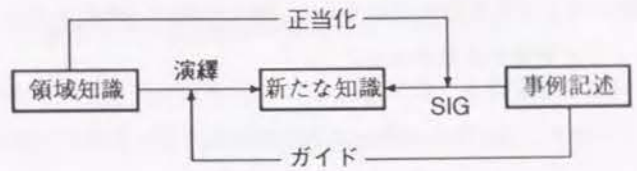


図 4.10: 知識獲得の正当化とガイドの関係

性質 1: SIG を正当化する能力をもつこと

性質 2: 事例記述を加えて設計知識となること

の 2 点が必要である。とくに, SIG を正当化する能力をもたせるためには, 予め完全な形で用意された領域知識が要求される。そのため,

問題点 1 予め与えられた知識から演繹しただけの知識が "新しい" 知識といえるか

問題点 2 予め完全な形で領域知識を用意することが現実的に可能なのか

という問題に遭遇する。

問題点 1 に関しては, 完全な (誤りのない唯一の説明を生成する) 領域知識を仮定し, 領域知識をできるだけ一般的に与えておき, 事例記述の説明に用いて特殊化することにより, 事例記述のガイドを必要としないという意味で "新しい", すなわち "合理的な" 知識を獲得する方法を 4.3.3節で示した。しかし, この方法では, つぎの点について考慮しなければならない。

初期領域知識ベースの構築 EBG で獲得された知識が真に新しい知識となるには, 初期領域知識は一般的なものを与える必要があるが, 一般的であればあるほど, 初期領域知識ベースを構築する際に, 論理的正当性が保てるように細心の注意を払う必要がある。

モジュール化された構造要素の利用 歪ゲージがひずみと電気抵抗効果を関係付ける部品である背後には, 弾性体の歪伝達効果, 抵抗が長さに比例し, 断面積に反比例するという法則などが, その機能達成を裏付けている。しかし一般に, 設計物の中で変換器として働く部品はモジュール化されており, また, その種類も限られたものであると考えられる。したがって, 各説明対象事例に対して, 逐一, 物理因果法則のレベルまで詳細化して説明づけるのは冗長であると考えられる。この場合, 機能空間から物理法則空間内の知識による根拠付けを待たず, 直接に, 構造空間と関連付ける初期領域知識の導入が有効である。

たとえば, つぎの領域知識を考える。第 1 引数は「構造素 X のもつ物理量 (またはその変化) N を検知する」ことを表す述語である。第 2 引数は「構造 X が L というデバイスの一種である」ことを表す述語である。

$$dt(transduce(X,N),[isa(X,L)]).$$

この知識を含意則として読むと, 任意のデバイス (L) によって任意の物理量 (N) が検知できることになる。これは明らかに over generalization (過度の一般化) されている。この知識を用いて事例を説明する場合, 物理法則, 物理効果の裏付けが無く, 一般に正しい説明を生成するとは限らない。すなわち, 機能と構造を連関させる一般的な知識には, 含意則として表現できないものが存在し, これを計算機上に記述するための新たな知識表現形式を考える必要がある。

4.5.2 類推過程正当化知識としての決定則と領域知識

2.5.2節に示したように, 決定則は類推プロセスを正当化するために提案された知識表現である。すなわち, 類推の正当化問題を SIG の正当化問題に帰着させ, どのような対象知識に対していかなる SIG を行なうことが正当化されるかの宣言, 換言すれば SIG において一般化することができる P と Q の関係の明示的な表現として, 決定則 $(P \succ Q)$ が導入されている。すなわち, 決定則は前節で示した領域知識に求められる**性質 1** を満たす。

ここで $P \rightarrow Q$ (P implies Q) ならば $P \succ Q$ (P determines Q) であるが, 逆は成立しない。すなわち一般則 $(P \rightarrow Q)$ は決定関係 $(P \succ Q)$ の特殊な場合として考えることができ, 決定関係をソース事例の情報で特殊化することで一つの一般則を獲得することができる。したがって, 決定則は前節で示した領域知識に求められる**性質 2** を満たす。

従来の EBG では, 含意則として与えられている領域知識を演繹的に組み合わせて新たな知識を生成する場合, 無意味な組み合せを排除するためのガイドとして事例が取り扱われている。すなわち獲得された知識の内容に事例記述が直接反映されてはいない。これに対し, 決定則を EBG 手法の初期領域知識として用いた場合, 初期領域知識から単純に演繹するだけでは知識を獲得することができず, 事例の情報は獲得された知識の内容を決める不可欠な要因として取り扱われる。このことは, 上記**問題点 1** を解決する。

また膨大な領域知識の下で, 「強い」含意関係知識を一つ一つ考えるよりも, 「Q を決定づける要因は何か?」のような形で考えることによって決定関係 $(P \succ Q)$ を抽出する方がより容易である。プログラマーは単に利用することができる決定関係を領域知識ベースに付け加えればよく, システムは SIG ができるときにそれらを用いる。これにより, 演繹推論のための知識は予め完全無欠の状態で用意しなくても, 事例の説明を繰り返

す度に妥当な (正当性の保証された) 知識が獲得され, その知識を領域知識に再格納しながら領域知識を整備するという方策が可能になる。すなわち,

1. プロダクションルールや含意則のような Q に対する厳密な充分条件よりも, Q の成否を決定する要因 (P) の方が簡単にコーディングすることができる。
2. 一つの決定則から事例記述を加えていくつもの含意則を生成することができる。

という点から, 決定則が利用できれば, 初期知識ベース構築が容易になることが期待できる。すなわち, 上記**問題点 2** を部分的に解決する。

4.5.3 決定則による知識獲得における一般化ルール

EBG と決定則からの知識獲得は, 何れも単一事例記述を一般化するという側面をもっている。しかし, 一般化を正当化する背景知識が, 前者では含意則であるのに対して, 後者では決定則という弱い規則であるという点に違いがある。このため, EBG と決定則からの知識獲得では, 事例記述の一般化 (identity elimination) において従う必要のある規則が異なっている。EBG は, 他の事例にも利用可能な形で知識獲得するので, 領域知識の中に変数 (X) と単一化される定数が存在すれば, そのときに限り変数 (X) が特殊化される。決定則からの知識獲得の場合はこの規則を用いることはできず, つぎの規則に従うことが決定則の定義から導ける。

自由変数の case set だけが一般化される。

すなわち, 変数 (X) が, 決定則 $P \succ Q$ において, P と Q に共有されないならば, そのときに限り変数 (X) は特殊化される。これを設計知識に応用すれば, predictor set が response set を明確に決定する関係を表現することができる。たとえば

$$construction(X,Y1), material(X,Y2) \succ elasticity(X,Z) \qquad (4.1)$$

は, 構造と材質によって, 弾性が決まるという知識である。ここで case set は $\{X\}$, predictor set は $\{Y1,Y2\}$, response set は $\{Z\}$ である。

4.5.4 決定則による知識獲得のアルゴリズム

4.3.2節で構築した EBG エンジンを拡張し, 上述の一般化規則にしたがって事例の一般化を行なう推論エンジンを, prolog (SICStus-prolog) を用いて構築した。

まず, X が変数でないときに限り X と Y を単一化する述語: sus_unif(X,Y) を定義した。厳密な十分条件だけではターゲット事例 (Ta) の説明ができなくなった時点における推論エンジンの動作を示す (図 4.11参照)。説明に失敗したノードを $q[t,Z]$ [2] とする。

[2] 大文字は変数, 小文字は定数または述語名を示す。また, 変数集合を引数とする述語は「述語名 [変数集合名 1, 変数集合名 2]」と表記する。

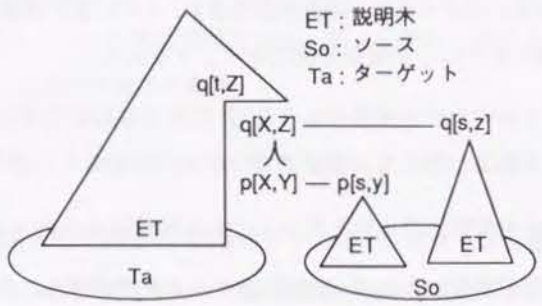


図 4.11: 決定則からの知識獲得アルゴリズム

1. 利用可能な決定則 ($p[X,Y] \succ q[X,Z]$) を探す。
2. $p[X,Y], q[X,Z]$ の複製 $p[X_1,Y_1], q[X_1,Z_1]$ を作成する。
3. $p[X_1,Y_1]$ をソース (So) と領域知識 (D) を用いて説明し, p を So に特化する ($p[s,y]$)。
4. $sus\_unif(q[s,Z_1], q[X,Z])$ により Z_1 と Z を単一化したあと, $q[s,Z_1]$ を So と D で説明して特化する ($q[s,z]$) と, 同時に Z も z に特化される。
5. $p[X,Y], q[X,z]$ の複製 $p[X_2,Y_2], q[X_2,z]$ を作成する。
6. $q[X_2,Z_2]$ と $q[s,z]$ とを単一化し, $p[s,Y_2]$ を得る。
7. $sus\_unif(p[s,Y_2], p[X,Y])$ により Y_2 と Y を単一化したあと, $p[s,Y_2]$ と $p[s,y]$ を単一化して Y_2 を y に特化すると同時に, Y も y に特化される。
8. $p[X,y], q[X,z]$ を得るので, 新たな領域知識 ($p[X,y] \rightarrow q[X,z]$) として登録する。
9. この領域知識を用いて, 一旦中断されていたターゲットの説明を再開する。

4.5.5 決定則の利用形態

説明木生成に決定則 $p \succ q$ を利用する場合, 標準的な EBG との唯一の違いは, q に対する陽な証明が必要なことである。q の証明に利用することができるのは, 領域知識 (D), 解析対象事例記述 ($Target : Ta$), 参考事例記述 ($Source : So$), および目標概念 (G) の 4 つであり, これらを組み合わせると合計 15 通りの証明方法が考えられる。説明木は, G を頂点として, D に含まれる含意則を用いてその十分条件を探索し, D, Ta, または So の記述と照合して終端する。したがって, 説明木中のノード q の証明に G を用いることができるのは $G \leftrightarrow q$ の場合に限られるため, 15 の組合せのうち, G を含む 8 通りは 1 つに集約される。また, Ta (および D) から証明可能ならば So を利用する必要はないことから, Ta と So がともに含まれる組合せは排除される。以上の考察より, 15 の組合せは以下に示す 3 通りに集約される。

[形態 1] D と Ta (またはその片方) から q が証明可能な場合 (図 4.12(a))
D と Ta (またはその片方) から G が証明できることとなり, 決定則を使う必要はない。これは, EBG を直接的用した場合に相当する。ただし, つぎの場合に限り決定則の利用が意味をもつ。

[形態 1'] Ta の中に q が含まれる場合 (図 4.12(b))
Ta に操作性規範を満たさない記述 ($q(t)$) が含まれることを許し, D と Ta から p が証明可能ならば, $p \succ q$ を使って操作性規範を満たす記述を得ることができる。

[形態 2] G と q が同値の場合 (図 4.12(c))
$G \leftrightarrow q$ となるのは, $G \leftrightarrow q \vee q_1 \vee \cdots \vee q_n$ かつ D, Tから $q_i (i = 1, \cdots, n)$ のいずれも導くことができない場合, ないしは $G = q$ の場合が考えられる。とくに後者は, 決定則 $p \succ q$ が設計物の基本プランとなる。

[形態 3] D と So (または So だけ) から q が証明可能な場合 (図 4.12(d))
機械機構のカタログやデバイスリストなどは, 機構名やデバイス名 (構造), その働き (機能) が, 仕組みとともに分類整理して記述されている。この形態は, これらの情報 (参考事例記述 So) を決定関係を用いて利用する。

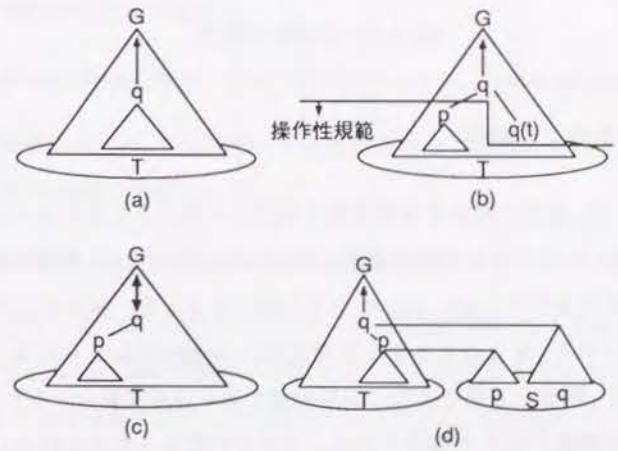
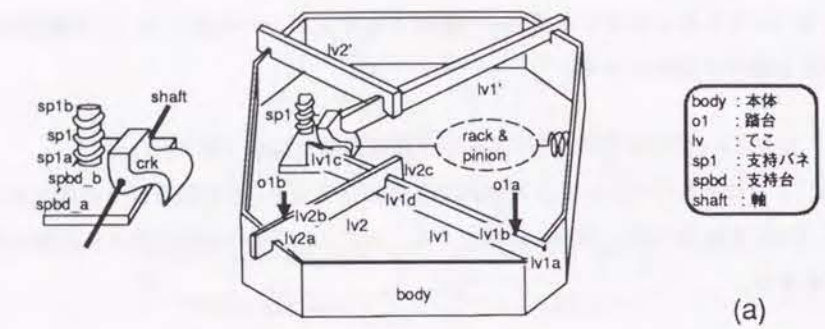


図 4.12: 決定則の利用形態



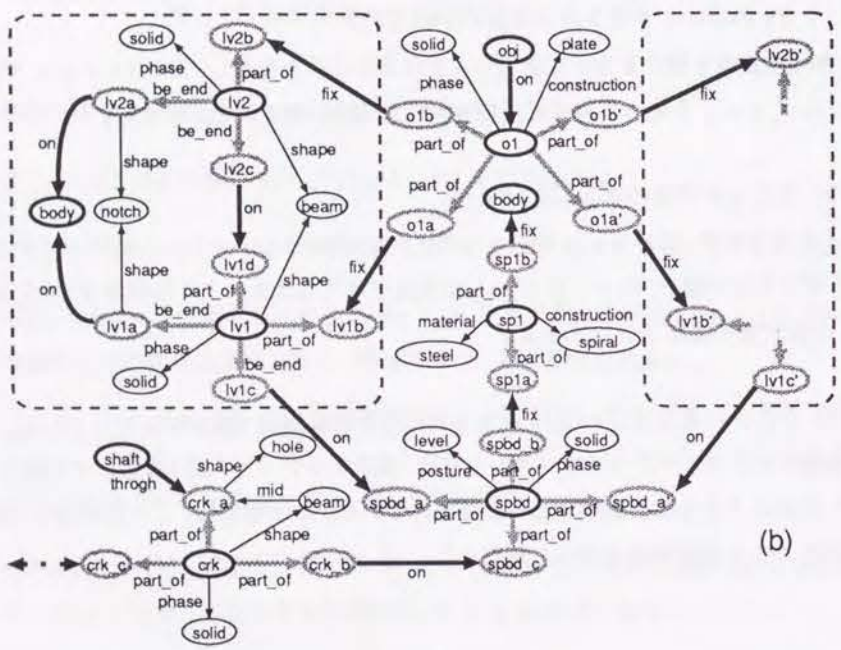


図 4.13: 体重計の構造

4.5.6 実験結果および考察

設計事例として, 体重に関連する物理量を体重計の踏台からクランクに伝達する機構をとりあげ, 提案したシステムの動作確認を行なった。図 4.13に, 事例の構造の概略と意味ネットワーク表現を示す。o1a, o1b の 2 点で踏台を支える一組のてこが, 図 4.13 の上下方向対象に 2 対あり, 計 4 点にかかる力が支持台 (spbd) に伝えられる。この機構を採用することにより, 踏台から受ける力と線形関係にある力が支持台にかかるとともに, 踏台の上下方向の移動量や傾きが抑えられる。支持台に伝わった力に関連した伸びが支持バネ (sp1) に生じ, それが支持台を上下方向に移動させる。その移動量は, クランク (crk) により水平方向の移動量に変換され, ラック・ピニオン機構により目盛板を回転させる。

まず, 体重計の構造情報 (図 4.13(b)) を 解析対象事例 (Ta) とし, その基本機能 ($measure\ (man, wight, meter, lotation)$) を目標概念 (G) としてシステムに入力する。これらの入力に対し, システムは 3 章で設定した機能・物理因果法則・構造に関する知識を用いて説明木を生成する。

図 4.14に示すのは, 全体の説明木の中から踏台 (o1) の垂直応力がクランクの位置変化を引き起こす説明の部分を抜き出したものである。体重計の説明には, 4.3.3節で設定した知識が用いられている。しかし, こういった強力な知識を予め不足なく用意することは現実的に不可能であり, この例ではつぎに示す決定則を使用している。各々の決定則から得られた知識を使って生成された説明木の部分を図 4.14に網かけ線で囲んで示す。

$$elasticity(X,Q) \prec construction(X,P_1) \wedge material(X,P_2) \tag{4.2}$$

$$measure(X,P,Y,Q) \prec detect(X,P,Z,R), trans(Z,R,W,S), transduce(W,S,Y,Q) \tag{4.3}$$

$$pp(X,P,Y,Q) \prec part\_of(X,Z) \wedge part\_of(Y,Z) \wedge be\_system(Z,R) \tag{4.4}$$

$$shape(X,P) \prec phase(X,Q) \tag{4.5}$$

(4.2) 式は, 説明木中のノード $elasticity(sp1, W)$ が, 解析対象事例 (Ta) や領域知識 (D) だけでは説明できなくなったときに形態 **3** で用いられる。実装システムでは, (4.2) 式から, 機械要素の規格表 [68] などの参考事例記述 (So) にあるつぎの記述

$$is\_a(jis\_g\_3512, spring) \wedge material(jis\_g\_3512, 'SWB') \wedge$$
$$construction(jis\_g\_3512, spiral) \wedge \cdots$$

と, 構造的に一般的に成立する規則 (D)

$$elasticity(spring, high), is\_a('SWB', steel), is\_a(steel, metal), \cdots$$

を加えて, 一旦特化したのち, 決定則の一般化規則に従って事例の名前 (jis_g_3512) を変数 Xに戻し, つぎの知識が生成される。

$$elasticity(X, high) \leftarrow material(X, metal), construction(X, spiral)$$

これは, 螺旋構造をした金属性の部品 (X) は一般に弾性が高いことを表している。これを用いて, 中断されていた説明木生成を再開する。この決定則を用いなければ, 全ての弾性体に対して各々弾性を有することの要因を人間が判断し, 領域知識ベースに記載しなければならない。

(4.3) 式は測定の基本プランであり, 機能間の決定関係である。システムは, これを形態 **2** で利用することによって, 図 4.14に示すように Ta の記述に特化した後, EBG の一般化規

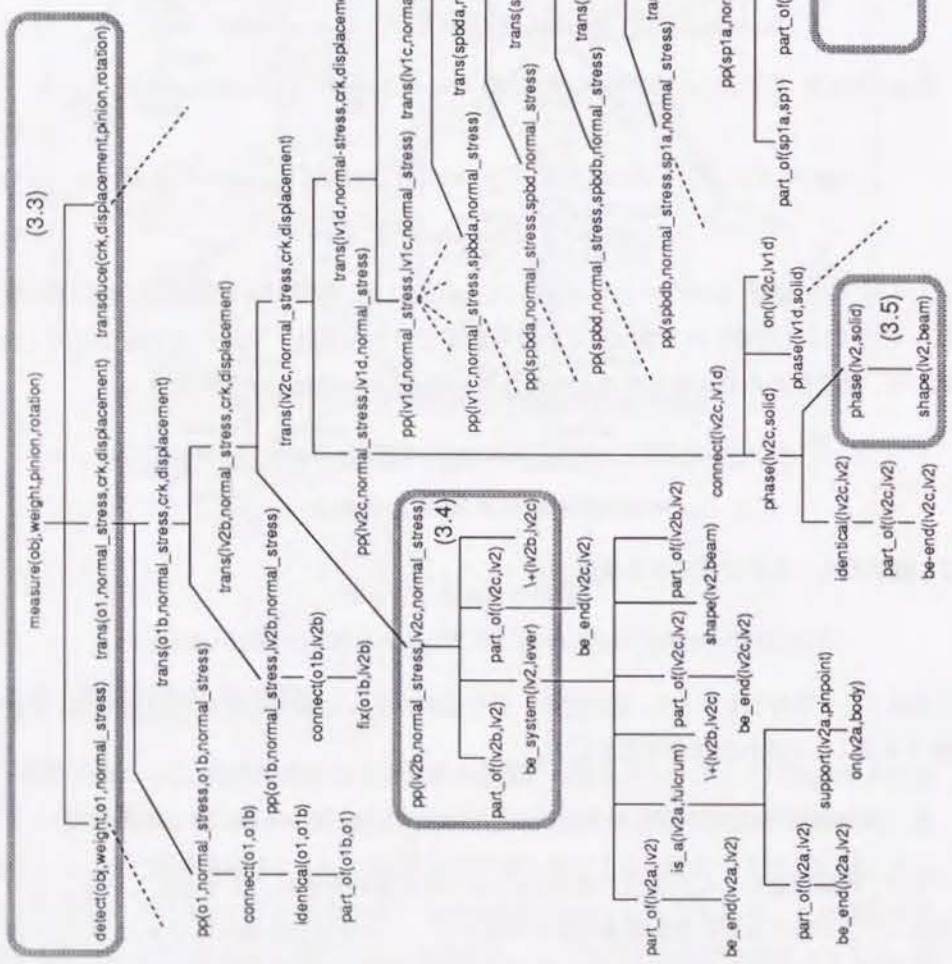


図 4.14: システムが生成した体重計の機能系統図

則に従って, X, Y, Z, Wを再び変数に戻し, その結果「測定対象 (X) の重さを部品 (Z) の垂直応力として取り込み (detect), それを部品 (W) の変位として伝え (transmit), さらに測定結果の利用者にとって利用しやすい情報である部品 (Y) の回転量に変換 (transduce) する」というプランを獲得する。

$$measure(X, weight, Y, rotation) \leftarrow$$
$$detect(X, weight, Z, normal\_stress) \wedge$$
$$trans(Z, normal\_stress, W, displacement) \wedge$$
$$transduce(W, displacement, Y, rotation)$$

(4.4) 式は,「X,Yという部品を含んでいる構造体 Zがどのようなシステム (R) であるかで, Xのもつ物理量 Pと Yのもつ物理量 Q との間に成立する物理因果関係が決まる」ことを表す機能–構造連関を示す決定関係である。これは, **形態 3** で含意則となり, 体重計の例の中で, 二点間の力の伝達の説明に利用されている。この決定則を使用しない場合, $lv2b$ から $lv2c$ までの細かな部品に注目し, それらの間の力の伝達を詳細に分析し, 説明構造を作らねばならない。これに対し決定則を用いる場合は, $lv2b$ から $lv2c$ までを一つのシステムとみなし, 力の伝達を「てこ」の動作によって説明している。すなわち, 決定則は, 詳細な解析を省略し, 説明構造の生成を一気に進める性質をもつ。また概念設計プロセスを, 目標機能 → 副機能 → 構造・属性 → 実体 を連続的に選択する行為としてモデル化した場合, 決定則から得られた設計知識は, 詳細な機能の分解を省略し, 一気に構造を想起する選択肢であるといえる。

(4.5) 式は,「Xの形状が Zであることが, Xの相が Yであることを決定する」ことを表す。これは構造間の決定関係であり, 形態 **3** で利用される。

4.5.7 決定則記述に対する制限

単一事例から複数の説明構造が生成されることは, EBG を応用するときにしばしば発生する問題である。領域知識が不完全であり複数の説明構造の各々を区別する情報が欠落しているか, または領域知識に誤りがある場合には, 複数の説明構造は互いに矛盾したものになる [69] [70]。しかし, 逆は必ずしも成り立たない。すなわち単一事例から複数の説明構造が生成されたとしても, それらが必ずしも互いに矛盾しているとは限らない。

決定則の適用範囲を広げるためには, 複数説明生成を許容する必要がある。もし, 複数説明の発生を領域知識が不完全である可能性を示す証拠とし, 単一の事例に対しては唯一の説明しかあり得ないという立場をとれば, 前節で示した形態 3 による決定則の利用ができなくなる。以下に, 形態 3 によって決定則を利用した場合, 図 4.15に示す 2 つの説明木 ET(1) と ET(2) が生成されることを示す。

決定則 $p[X,Y] \succ q[X,Z]$ を用いるとき, ソース事例の $q[X,Z]$ 達成の説明木 (図 4.15 の ET(1)) に $p[X,Y]$ の説明木全体が含まれることはない。なぜなら決定則を使うまでもなく含意則で $p[X,Y] \rightarrow q[X,Z]$ が演繹されることを意味するからである。ところが, 一旦, 決定則からの知識獲得により $p[X,Y] \succ q[X,Z]$ から $p[X,y] \rightarrow q[X,z]$ が導かれると, これを用いた $q[X,Z]$ に対する説明木 (図 4.15の ET(2)) は $p[X,Y]$ に対する説木を含む。したがって, 少なくとも 2 つの説明木, ET(1), ET(2) が生成される。

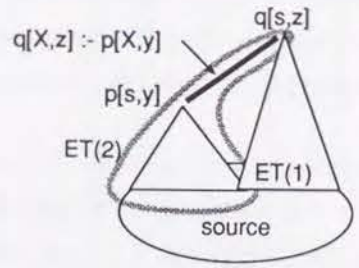


図 4.15: 複数説明の発生

本研究では, 事例の記述は, 設計図面などから機械的に, かつ, 機能的な考察なしに容易にコーディングできること, および定性物理の NFIS(no function in structure) 原理を考慮して, 結合形態, 形状, 材質, などの表層的な構造情報に限定している。そのため, 形態 3 が利用できない場合には, 機能–物理法則間, 物理法則–構造属性間や機能–構造間の決定関係などが利用できなくなる。

以上の考察より, 決定則を利用するためには, 積極的に複数説明生成を許容する必要があると結論される。

4.6 獲得設計知識の評価

4.5節では, 決定則を EBG の領域知識として用いることによって, 決定則を事例情報により特化する形で設計知識を獲得した。これを “決定則からの知識” と呼ぶ。また, 4.4 節では, 決定則からの知識および予め含意則として与えられている領域知識から説明木を生成ならびに一般化し, その任意の部分木を 5 つのモードで抽出し, 部分木の末端にある機能／構造の連言が, 部分木の根にあたる機能／構造に対する十分条件とする設計知識を獲得した。これを以下では “説明木からの知識” と呼ぶ。ここで, 獲得された知識 (説明木からの知識と決定則からの知識) を, あらかじめ与えられていた知識 (初期領域知識および決定則) と比較検討する [4][67]。

これらの知識を設計知識として, 設計フェーズへの適合性, 知識利用の合理性 (rationality), 設計プロセス進展の効果性 (efficiency) の 3 つの視点から評価する。とくに後者

の二つの視点から, 各々の設計知識は図 4.16に示すように整理することができる。

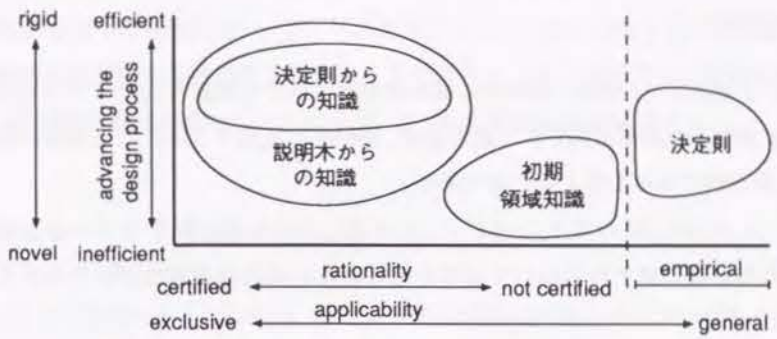


図 4.16: 二つの尺度からの設計知識の評価

4.6.1 適合性

どのような記述の設計知識が操作的であるかは, その知識を必要としているシステムによって変化する。すなわち, 2.4.1節に示したように, 設計局面や設計者の意図によって必要とされる知識の質が異なってくる。まずこのような知識の質, すなわち設計知識の適合性の側面から獲得された知識を評価する。

提案した手法では, 説明木上の注目するノードに獲得したい目標を設定し, 必要となる空間を説明レベルに指定することによって, それぞれの設計フェーズに適合した知識を抽出できることを 4.4節で示した。

モード I による知識は, 機能構成段階への適合性が高く, II, III による知識は, 基本機能のような幾つかの副機能に展開される機能から一気にそれを実現する構造にまでブレークダウンする段階への適合性が高い。また, IV よる知識は機能–物理法則–構造・属性という概念設計知識モジュールであり, 個々の機能を構造に対応付ける段階への適合性が高い。V による知識は常套的に用いられる構造説明モジュールを表わしており, 構造実現段階への適合性が高い。

II と III を比較すると, 同じ適合性を有しているが, II で獲得された知識の方がより一般性が高く, III では設計案を進展させる効果性が高い。一般に, 一つの説明レベル (空間) にあるノード群は階層を構成しており, ある目標概念に対する説明としてどの階層のノードを用いるかに自由度がある。この自由度を利用し幾種類かの説明を生成し, EBG 法を複数例題の場合に拡張する手法も提案されており [71], 説明ノードの選択は EBG 法を適用する際の重要な観点である。モード III では説明を一般機能系統図の末葉だけの連言に限定 (指定) し, 事例記述に即した構造的記述による説明を得ている。しかし, できるだけ説明木の根 (目標概念ノード) に近いノードを選択し, 説明の一般性を向上させることが

できる [48]。モード II は空間 L との操作性境界に最も近いノード, すなわち構造的記述の範囲で最も根に近いノードを選択し, 最も一般的な構造的説明を抽出するモードである。

4.6.2 合理性

知識の合理性とは, 知識を機械的に組み合わせて新たな設計を行なうプロセスにおいて, 知識を適用した結果生成された設計物が, 物理的に実現できることが裏付けられている知識を操作的であると捉える立場である。

モード I, II, III, IV のように, GC と LE の間に媒介知識空間 P あるいは L が介在する場合, 獲得知識にはそれに応じて目標合理性あるいは因果合理性が付与されることになる。

本システムでは空間 L と空間 P を予めコンパイルすることを避け, 空間 P には因果合理性を前提としない一般的な可能性を示すプラン型知識を格納しておき, 説明木生成に際して, 多数の事例解析を通して, これに合理性を与える。したがって, 初期領域知識を任意に組み合わせた知識は, 必ずしも設計に有効な知識である保証はないが, 説明木から得られた知識は, 合理性が与えられているためこれを保証する。したがって, 6章で後述するような, 知識を機械的に組み合わせて新たな設計を行なうプロセスにおいては, 生成された設計解が物理的に実現できるかどうかは, 後者の知識を利用した方が保証されている。図 4.16では, この保証の程度を横軸で表している。

ところが逆に, これらの知識が利用できる自由度を考えると, 説明木からの知識は, 解析対象事例のガイドにより領域知識をいくつか組み合わせたものであるから, 初期領域知識とくらべると, 解析対象事例に類似の設計に利用範囲が限定される。

決定則は, 2.5.2節 に述べたように, 事例情報が加えられてはじめて演繹推論に利用することが可能になる。すなわち決定則は初期領域知識のような理論的な記述ではなく, 過去の経験によって得られたバイアス的知識であるといえる。したがって合理性は保証されていないが, 多くの設計局面に適用を試みることができる。ただし, 一旦事例記述を加えて含意則にすれば, 説明木からの知識と本質的には等しくなる。

以上の関係を, 図 4.16の横軸で示している。

4.6.3 効果性

設計プロセスを, "目標とする基本機能が与えられると, それを詳細な副機能列に展開し, つぎに各々の副機能を達成することのできる構造属性をもつ物理的実体を生成するプロセス" であると捉えると, 設計知識はこの設計プロセスを進展させる度合, すなわち設計知識の**効果性**で評価することができる。効果性は, 概略的に見て, GC と LE の間の隔たりに対応する。提案した手法では, 一般化された説明木から抽出する部分木の大きさによって, 進展の度合の大きな知識から小さな知識まで, 任意に獲得することができる。

II, III の形式の知識は, ある基本機能の達成に必要な構造を設計するのに適用される。III の形式の知識によれば II より具体的な構造まで一度に決定される。したがって, III, II, I の順に適用時の効果が高い。IV, V の形式を III の形式と比較すると, より抽象的な機能・構造から具体的な構造を決定することができるという意味では, 適用時の効果は III, IV, V の順に効果が高い。したがって, 以下の相対的関係が成立する。

$$III > IV > V$$
$$III > II > I$$

このような説明木からの知識は, いくつもの初期領域知識をチャンク化した知識であるため, プロセスを進展させる度合は, チャンクされる前の個々の初期領域知識より小さくなることはない。

経験則的な側面をもつ決定則は, ある機能を達成するための構造的制約条件と, その背後にある物理因果関係の説明を省略して直接関係付ける。また, 4.5.5節でも述べたように, 決定則から得られた知識は, 詳細な分析を省略し一気にプロセスを進展させる知識であるといえる。このような性質をもつ知識は, 要求機能からそれを実現する構造を発見するプロセスを進める度合が大きい (効果的である) 反面, 小さなステップをきざむ初期領域知識と較べると, 組合せの自由度が小さく, 新奇な設計解を得る可能性が小さいと考えられる。すなわち, 新奇なアイデアを導入する余地の少ない固定的な側面をもつ。

以上の関係を, 図 4.16の縦軸で示している。

4.7 結論

本章では, 概念レベルでの設計支援のための知識獲得システムを提案した。これは, 2 章で示した階層的設計知識を用いて設計事例を解析し, 複数の領域知識と設計事例のもつ情報とを加えて 1 つの新たな知識にコンパイルする手法を用いている。これはまた, 演繹推論による知識生成と, 単一事例からの一般化手法 (SIG) の融合した手法であるといえる。しかし, そこには知識獲得のための知識獲得というパラドクスや, 生成された知識が真の意味での新しい知識ではないという問題が生じる。

本研究では, これらの問題の解決策として, 領域知識に, 事例解析情報が加えられて初めて因果合理性が得られるように一般性をもたせることを提案した。この枠組みでは領域知識は本来妥当性をもたない SIG を正当化するために使われており, 過大な一般性を与えてこの能力が失われないように注意する必要がある。そこで, 含意命題形式で与えられている領域知識に対し, より収集が容易で, より一般性をもった記述形式である決定則を導入し, 論理的な正当性を失わずに, 一般化された知識から設計知識を獲得する手法を提案した。

また，階層構造的に整理された設計知識を用いて説明を行なうことにより，説明木上に操作性境界が明確に設定可能となる。これによって，一つの事例から多面的な視点の下で種々の操作性知識を抽出するモードの設定が可能となった。

最後に，獲得される知識の質をこの獲得モードとの関連で明らかにした。

第 5 章

設計物からの設計意図解析と知識抽出

5.1 緒言

本章では，公理論的設計アプローチに基づいて既存の設計物を解析し，「なぜ設計者は基本機能達成にこのような構造を用いたのか」(Why explanation) についての知見を得る手法を提案する。本章では，これを深い説明と呼ぶ。

4 章で提案した知識獲得手法は，解析対象事例が「どのようにして基本機能を達成しているか」の説明 (How explanation) である機能系統図を作成し，そこから設計知識を抽出する。抽出された設計知識は，基本機能実現のための十分条件である。すなわち，目標とする機能を達成するための設計知識が複数個利用可能な場合，原理的にはそのいずれを使ってもよい。しかし，実際の設計では様々な条件を考慮して，多くの選択枝の中から目標の達成に使用する手段が必然的に決定されている。その決定には，「なぜ」そのような手段を用いるかを説明する方略的 (メタプラン的) な知識が用いられていると考えることができる。

すなわち，注目する設計物が「なぜ」そのような下位機能群，物理因果連鎖，および構造属性を用いて実現されたのかを解析することは，設計知識を利用するメタプラン的な知識を得るために不可欠なものである。

5.2節で提案する手法は，解析対象事例が公理論的設計アプローチにおいて提案された二つの設計公理を満たす構造をもつことを前提とし，一旦単一の機能だけを達成するように構造を最小化することによって他の機能との干渉を顕在化したのち，構造付加により干渉の回避を非単調に行なうプロセスの履歴として，メタプラン的な設計知識を導出する。

5.3節では，このプロセスを，物理因果連鎖に注目した設計物の構造変更，および TMS (真実性維持機構) を用いた設計物記述を導入することによって，システマティックに実現する方法を提案する。

5.2 設計意図解析方法

4 章で提案した EBG システムによって生成した説明木は, 既存の設計事例において目標概念 (基本機能) が, 下位機能群, 物理因果連鎖および構造・属性の利用によって「どのように (**how**)」達成されているかを説明付けるものである。この説明木から抽出される知識は, 基本機能実現のための十分条件である。すなわち, 目標達成のための設計知識がいくつも利用可能な場合, そのいずれを使ってもよく, 同じ機能を達成する構造 (設計解) は複数個生成される。しかし, 実際の設計では様々な制約条件を考慮することにより, 多くの手段の中から使用する手段が必然的に決定されている。

そこで, その設計物が「なぜ (**why**)」そのような下位機能群, 物理因果連鎖および構造・属性を用いて実現されたのかを説明付けるために, EBG システムによって生成された説明構造を, 設計プロセスを良い方向に導く原理 (設計公理) に沿って調べる方法を提案する [12][72]。

このような説明を導くために, 基本機能である「流速の測定」と, たとえば「高温に耐える」, 「局所的な流速測定を行う」といった他の機能との間の物理空間上でのインタラクションに注目する。これによって, 基本機能に対する一般機能系統図中の説明には用いられていない構造・属性の必要性を示す「深い」説明を, 公理論的設計アプローチに基づいた複数の機能の干渉と回避の履歴として導き出すことができる。

5.2.1 公理論的設計アプローチ

2.3.2節で述べたように, Suh らは, 公理論的設計アプローチにおいて, 設計活動全般において成立し, 設計者を良い設計に導いてくれる原理・原則, すなわち設計分野における公理としてつぎの 2 つを提案している。

公理 1(Independence Axiom) : 機能的要求の独立性を維持せよ

公理 2(Information Axiom) : 情報量を最小にせよ

公理 1 は, 互いに独立に定義された機能的要求が設計解においても独立性を維持されていれば, その設計は良い設計であることを保証している。一方, 公理 2 は, 公理 1 を満足する設計解のうちで情報量が最小のものが最良の設計であることを保証するものである。

実際の設計プロセスには, 公理 1 (機能の独立性の維持), 公理 2 (情報量の最小化) がそのまま適用されることは少ない。その代わり, 2.3.2節で示した具体的な規則として記述された系が, 設計プロセスを導く働きをする。これらの系は二つの公理をコンパイルして操作可能にしたものである。

本研究では深い説明構造の導出に二つの公理を直接用いて, メタプラン知識を導き出す。導入された知識は, 公理をコンパイルして操作可能な知識にしたもの (系) であると

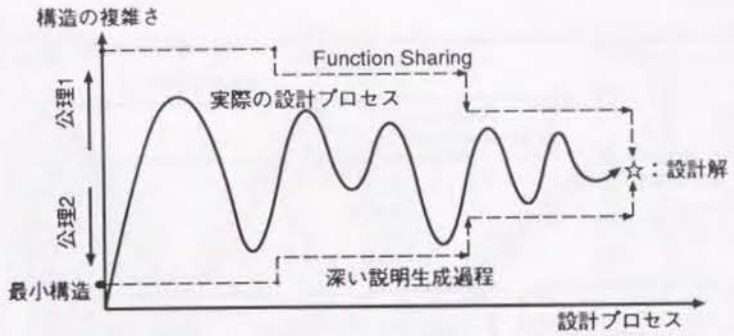


図 5.1: 設計プロセスと構造的複雑さ

みなすことができる。

5.2.2 深い説明生成

実際の設計物は試行錯誤の末に 2 つの公理を満たす構造, すなわち機能の干渉がない上で最も情報量の少ない設計解を得ていることを前提として考える。この前提に立つと, つぎに示すプロセスで, 設計者がなぜこのような設計解に至ったか (深い説明) を, 機能間の干渉回避という側面から解析することできる。

まず解析対象事例の構造記述の中から, 最も基本的な機能を満たすための構造・属性を抜き出す。これを最小構造と呼ぶ。設計公理 2 に従えば, 機能を達成するために不必要な構造属性は情報量を無駄に増やすだけであるから設計案に含まれていない。すなわち, ここで排除された構造・属性は, 他の機能を満たすために存在していると考えられる。したがって, 最小構造では達成できない機能が存在することになる。達成できない機能は, 先に示した前提により, 構造を付加することにより回復することができる。この構造付加を繰り返すことによって, すべての要求機能を満足する設計解に至る。

ここで示したプロセスは, 実際の設計プロセスそのままをトレースするものではない。両者は, 概念的に図 5.1に示す関係にあると捉えることができる。

設計解は, 公理 1 に従えば構造を複雑化する方向へ, 公理 2 では逆に簡略化する方向へ導かれる。実際の設計プロセスは, 簡略化と複雑化を繰り返し, 試行錯誤的に進行すると考えられる。計算機設計プロセスのモデルとして提案された Function Sharing [24] は, 複数の機能的要求に対して, それぞれを達成する構造を個別に設計し, それらをまとめた構造体, すなわち無意味な構造をもたない内で最も複雑な構造体を作成し, これを公理 2 に従って構造的に簡略化する手法であると解釈することができる。一方, 本章で提案する深い説明生成過程は, 3 章で提案した手法の一般化能力を利用して最小構造を作成し, これを公理 2 に従って全ての機能を満たすように構造・属性を付加する手法であるとい

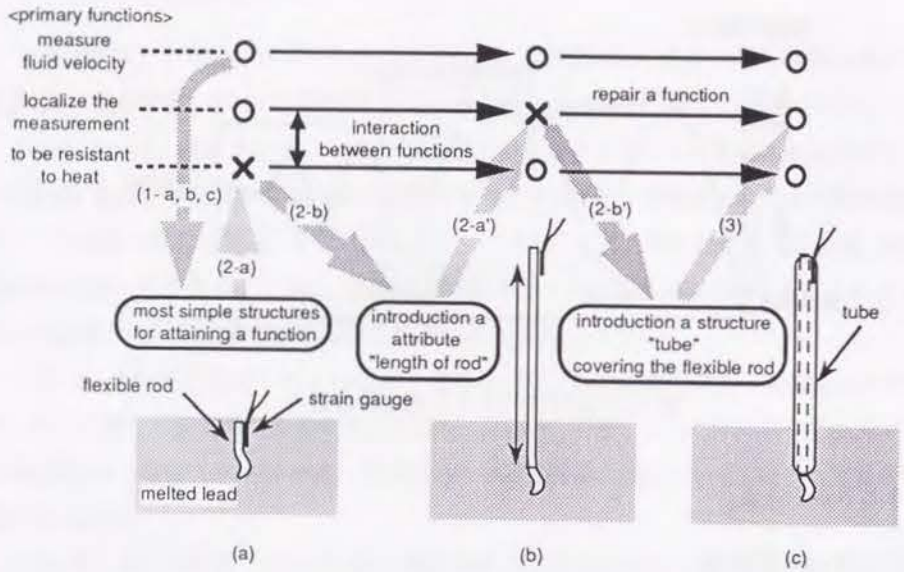


図 5.2: 可撓棒の深い説明

える。

(1) **最小構造の導出**　最も基本的な機能的要求 (GC; Goal concept) だけを達成するための最小の属性の組をもつ最も単純な構造 (単純化された設計案) を, 以下に示す (1-a) から (1-c) のプロセスを用いて求める。可撓棒の例では, GC は「流速の測定」であり, 結果として求まる**最小構造**は, 図 5.2(a) のようなものとなる。流速を歪ゲージに伝える棒の「長さ」という属性は, 使用しなくても基本機能を達成することができるため除去されている。

最小構造導出段階は, 詳細な制約を考慮せず基本的な設計を行う段階に相当し, 公理 2 に基づいて, 目標達成のための手段として最も簡単な構造を選択することに対応する。

(1-a) 一般化構造の導出　ある単一の GC (基本機能) に対する機能系統図を生成し, GC を達成するための構造・属性を表す知識を, 図 4.7に示した抽出モード III, すなわち機能系統図の末葉の連言として抽出する。可撓棒の場合, GC は「(一般に) 流速を測定する」であり, 抽出された知識は, EBG 手法のもつ irrelevant feature elimination の働きによって, 図 3.1の意味ネットワーク中に含まれる構造属性の中から, GC に無関係な特質はフィルタリングされて取り除かれている。また, もとの構造には陽に含まれてはいないが GC に関連のある特質が領域知識から付加され, その結果の意味ネットワークが図 5.3(a) に示すものである。

つぎに, 機能系統図を identity elimination によって一般化するとともに, 設計知識空

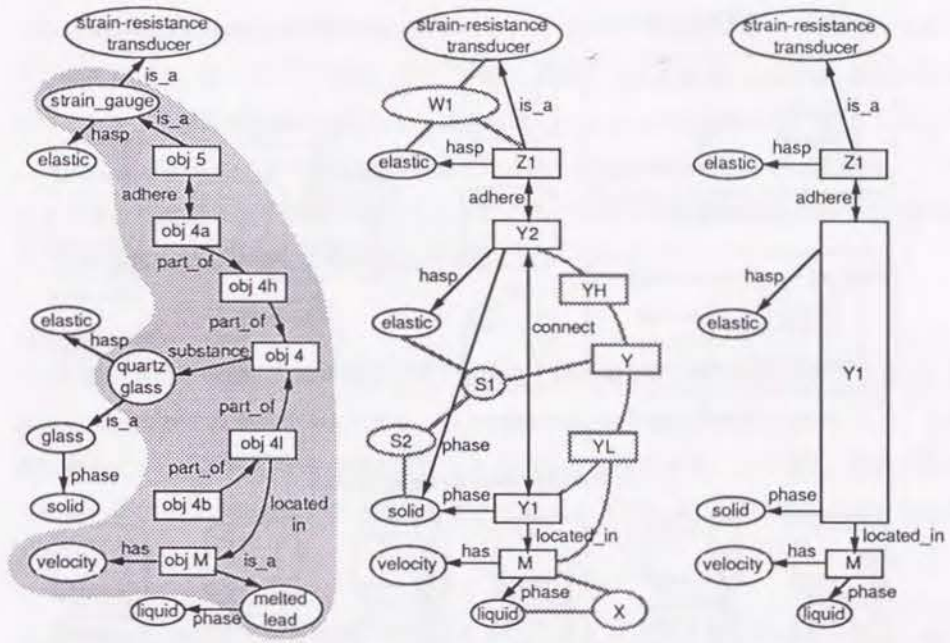


図 5.3: 意味ネットワークの最小化操作

間 S から生成されたノードと空間 L から生成されたノードとの境界まで機能系統図を刈り込み (operationality pruning), 一般化レベルを上げる。これは図 4.7のモード II で知識を抽出することに相当し, その結果として生成される最も一般化 (抽象化) された意味ネットワークは, 図 5.3 (b) に示されたものであり, 流速測定一般に有効な構造を与えている。

(1-b) 構造の最小化　公理 2 に従うと, 設計物に含まれる情報量は少ない方がよい。したがって, (1-a) で抽出された構造・属性の中から冗長なものを発見し, 除外する必要がある。この冗長性排除は, 物理因果連鎖に注目して行なうことができる。すなわち, **物理因果連鎖の統合則**を考え, 注目している物理因果連鎖に適用して最小化する。

3.3.4 節で導入した物理因果関係の一般型を, 再び図 5.4に示す。ここでは便宜上 TYPE I を P1≠P2 の場合 (AB 型) と P1=P2 の場合 (A 型) に分け, TYPE II を B 型と表記する。3 つの型 (AB, A, B) の中の二つを直列に組合せると, 図 5.4 に示す 6 通りの物理因果連鎖統合パターンを考えることができる。その内, B+B=B, B+A=AB, B+AB=AB は構造上の変化をもたらさない。これらは, いくつかの因果連鎖をマクロ的に 1 つの因

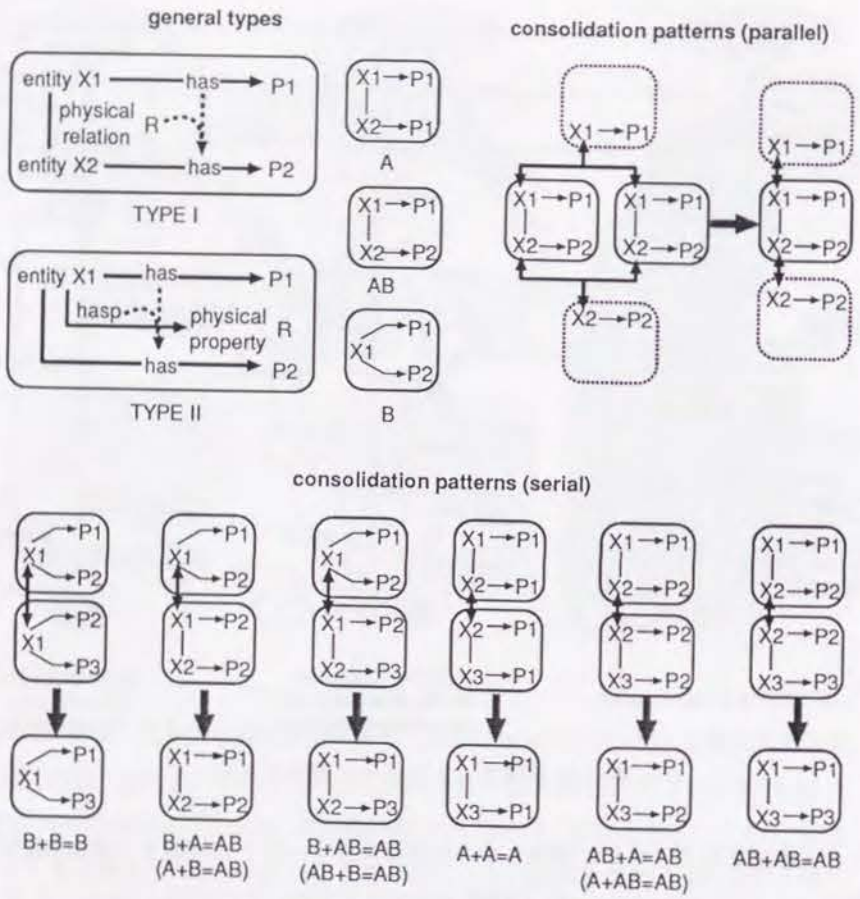


図 5.4: 因果連鎖の統合パターン

果関係と把握することであり, 物理因果連鎖の粒度を大きくするに過ぎない。残りの 3 つは, 構造上の変化をもたらすため最小構造の導出に寄与する。ただし, AB+AB の結果は, 必ずしも因果関係が把握することができるとは限らない。概括的にいうと, たとえば, ピストン (X1) の上下運動 (P1) がスクリュー (X2) の回転 (P2) を引きおこし, それがボディー (X3) の推進力 (P3) を引き起こす因果連鎖を考えた場合, これを統合するとピストンの上下運動とボディーの推進力との間に直接的な因果関係を与えてしまうことになる。以上の考察から, 最小構造の導出に利用することができる統合パターンは, A+A=A および AB+A=AB であると考えることができる。また, 並列の統合パターンとしては, A・A=A, B・B=B, および図 5.4に示す AB・AB=AB の 3 つが考えられる。

可撓棒の例では, 3 章で示した図 3.5(a) に示す物理因果関係 L1, L2 に対して AB+B=AB なる統合パターンが適用され, 結果は図 3.5(b) となる。

(1-c) 最後に, 物理因果連鎖の統合に伴う構造・属性の単純化を行なう。図 3.5 に示した統合の結果, Y2 が Y1 に単一化されている。すなわち, 流速測定という機能の達成だけを考えた場合, 歪みゲージを貼る部分 (Y2) と液体に浸す部分 (Y1) が異なる必然性が無いことが, 物理因果連鎖の統合により導き出されている。物理因果連鎖が成り立つために Y1 と Y2 がもつべきであった構造属性は, 単一の実体 Y1 がもつことになる。このような構造的変更の結果が図 5.3(c) である。これに相当する設計物の具体的な形 (構造) は, すでに図 5.2(a) に示した**最小構造**である。

(2) 構造付加

(2-a) 機能達成の確認 1 つの機能 (GC) を満たす最小構造が, 他の機能的要求 (GC's) を満足しているかどうかをチェックする。この例題では, 他の機能的要求 (GC's) とは, 「高温に耐える」, 「局所的な流速測定を行う」といった要求である。この場合, 高温に耐えられない歪ゲージが熔融鉛の高温にさらされていることから, 前者が満足されていないことが示される。

(2-b) 構造付加による機能回復 GC's のうちいずれかが満足されていないなら, その GC's を支持するような付随的構造および属性を探索する。例題の場合, 「高温に耐える」という GC's を回復するものとして, ロッドのもつ「長さ」という属性が見い出される。

(2-a') 良い設計では, 構造上の一体化をさせながら機能的独立の独立性を保つ場合が多い。そのような場合には, 一箇所の変更が他の部分に影響を及ぼし, その時点までのメタプランによる工夫によって保たれていた機能の独立性が崩れてしまうことがある。すなわち, ある機能的要求を満足させることによって, 他のすでに満たされている機能が満たされなくなることが起こる可能性がある。そこで, 再び (2-a) を繰り返してチェックする必要がある。

実際, この例題では, (2-b) で加えた「ロッドの長さ」という構造属性のために, ロッドの検出部以外の部分も流体抵抗を受けてしまい, 図 5.2(b) に示すように, 「局所的な流速測定を行う」機能が満たされなくなる。

(2-b') (2-a') で発見された干渉を解消し, すべての機能を達成するため, 新たな構造・属性が加えられる。これは, はじめに選択した目標達成手段が公理に照らして適当ではなかったために, 別の手段を探すことを意味している。このような, 目標達成のための手段選択の合理的な試行錯誤が設計時にも行われていると考えられる。この際に判断基準となるのは設計公理 1 である。ここでは, 実際に試行錯誤を行うのではなく, 既に設計が終わったものから知識を獲得する。したがって, 新たな構造・属性は入力事例の構造・属性のうち未使用のものから探索される。

そこで, 図 3.1の意味ネットワーク中から, 満足されなくなった GC's を修復するための部分構造あるいは属性を見いだすと, この例では, 図 5.2(c) に示すように, ロッドを被う筒管を設計案に付加することによって「局所的な流速測定を行う」という GC を回復する。

(3) 終了 すべての機能が達成されるまで (2-a) と (2-b) を繰り返す。これらの一連の説明の履歴として, 図 5.2(a),(b),(c) に示す測定器の深い説明構造が導き出される。

5.3 深い説明のシステマティックな生成

5.2節で示した解析手法において, 複数機能間の干渉, すなわち, ある機能を達成するために導入した構造属性による他の基本機能達成の妨害が生じる。これを回避するために新たな構造属性を導入すると, それは別の干渉を引き起こす可能性もある。すなわち, 干渉回避のプロセスにおいては, 構造属性の導入にともなって, それによって達成される機能は必ずしも単調に増加するのではなく, 場合によっては減少する (非単調である)。そこで, 機能・物理因果法則・構造を記述し, 干渉回避プロセスの進行にともなって新たな干渉の発見をシステマティックに行なう手段として, 2.5.3 節で説明した「TMS(Truth Maintenance System : 真実性維持システム)」を用いる。

TMS を用いた深い説明構造生成を説明するために, 体重計を例題として用いる。体重計の場合, 最も基本的な要求機能「体重を計る」の他に,「踏台を水平に保つ」, 体重と目盛の回転量との間の線形性を与えるために「重さを一点に集める」ことなどが要求される。本例の場合,「体重を計る」という基本機能に対する機能系統図の一部は図 4.14であり, 述語名 pp で与えられるノードは, 機能達成を説明付ける物理因果連鎖 (図 5.5) の L5 から L15 までの部分に対応している。図 5.5に示すように, 本例は, 接合し合う部品の片方に生じた垂直応力が他方の部品にも垂直応力を引き起こす因果関係 (M1), 接合する固体の一方の移動が他方の移動を引き起こす因果関係 (M2), テコの一端に生じた垂直応力が他端にも垂直応力を引き起こす因果関係 (M3), および弾性体に生じた応力は弾性体そのものの変位を引き起こす因果関係 (M4), を用いて体重を計る機能を実現している。

準備段階

TMS が生成するノードを, **構造素の属性**を示す s_n, **構造素間の関係**を示す r_n, **物理法則**を示す L_n, **部分機能**を示す F_n, **基本機能**を示す G_nとする。各々のノードには In 状態または Out 状態が割り当てられる。その状態が他のノードの状態に依存する場合, その依存関係をあらわす justification(正当化) を当該ノードに与える。J_nを justification の

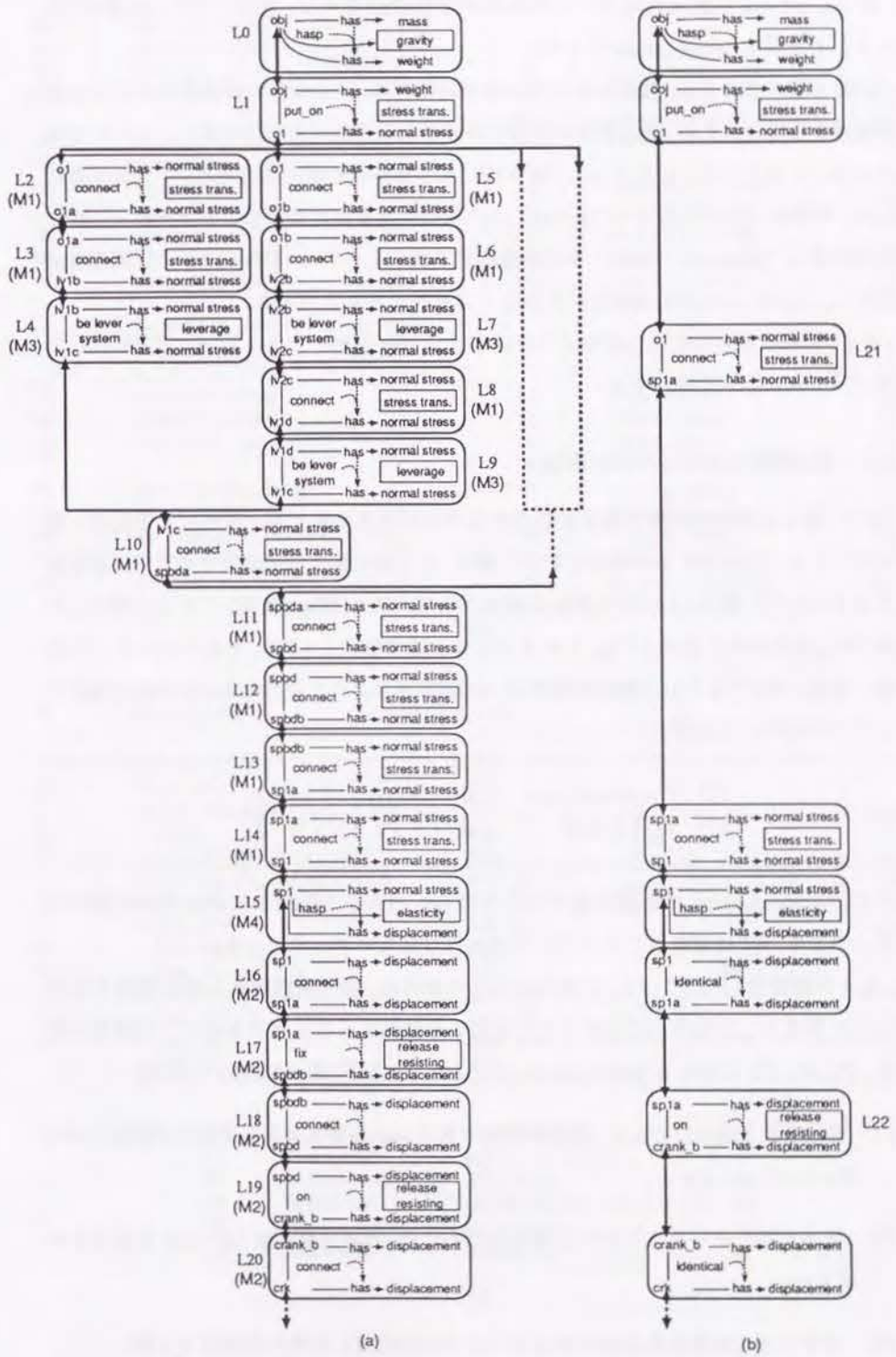


図 5.5: 体重計の物理因果連鎖 (部分)

番号とし, $sl([INlist],[OUTlist])$ [1] を, $INlist$ に含まれるノードの状態が全て In であり, かつ $OUTlist$ に含まれるノードの状態が全て Out ならば, 当該ノードの状態が In になることを示す justification とする。

3.5節で述べたように, 解析対象事例の表層的構造は, いくつかの構造属性で記述された構造素を示すノードが, 構造素間の関係を示すアークで結ばれた意味ネットワークで表現される。これを TMS に与えると, 図 4.13に示す体重計の構造情報から, たとえば測定物 (obj) が踏台 $(o1)$ の上にある "$on(obj, o1)$" 等の構造素間の関係を表すノード r_n, 踏台が固体である "$phase(o1, solid)$" 等の構造属性を表すノード s_n が生成される。図 5.6(a) に示す s_n, r_n は, この時に TMS が生成したノードのリストである。

まず, 初期状態としてこれらのノードはすべて Out 状態としたのち, 5.2節で提案したアルゴリズムに従って推論する。

(1-a) : 基本機能に対する一般化構造

まず, 最も基本的な機能的要求を達成するための構造を考える。例として体重計の重さをクランクに伝えるまでの機構を考えた場合, 最も基本的な要求機能である「体重を測定する $(G1)$」は, 図 4.14 に示す機能系統図からモード I (図 4.7参照) によって抽出した知識から, 体重を取り込み $(F1)$, それをクランクまで伝達し $(F2)$, そこでピニオンの回転量に変換 $(F3)$ するという副機能群によって達成されることが分かる。これを TMS のノードでつぎのように表す。

$\overline{G1}$	$Contradiction$	:	$sl([\,],[G1])$	$J1$
$G1$	体重を測定	:	$sl([F1, F2, F3],[\,])$	$J1'$

ここに, $Contradiction$ は矛盾を表すノードであり, TMS ではつねに Out 状態に維持される。つまり, $G1$ は常に信じられていなければならないノードである。

各々の副機能 $(F1, F2, F3)$ を満たすための条件は, 機能系統図から当該機能を表すノードを頂点とした部分木に注目することにより抽出することができる。この構造的条件を, $F1, F2, F3$ に対する justification$(J11, J12, J13)$(図 5.6(b) 参照) で表す。

F1　体重を取り込むためには, 踏台が固体であり $(s1_3)$, 質量のある測定物が踏台に乗っていれば $(r1)$ よい。

F2　重さを踏台からクランクに伝達するためには, 物理因果連鎖 L5～L20 が成立すればよい。

F3　クランクの変移を目盛盤であるピニオンの回転量に変換する部分は省略。

[1] prolog を用いて TMS を計算機上に実装したため, 本章では justification を節形式で記述する。

ノード		ノードの示す内容		$justification$
$s1_1$	:	$part\_of(o1_a, o1)$	:	$[\,]$
$s1_2$	:	$part\_of(o1_b, o1)$	:	$[\,]$
$s1_3$	:	$phase(o1, solid)$	:	$[\,]$
$s1_4$	:	$construction(o1, plate)$	:	$[\,]$
$s2_1$	:	$be\_end(lv1_a, lv1)$	:	$[\,]$
$s2_2$	:	$part\_of(lv1_b, lv1)$	:	$[\,]$
$s2_3$	:	$be\_end(lv1_c, lv1)$	:	$[\,]$
$s2'_3$	:	$part\_of(lv1_d, lv1)$	:	$[\,]$
$s2_4$	:	$phase(lv1, solid)$	:	$[\,]$
$s2_5$	:	$shape(lv1, beam)$	:	$[\,]$
$s2_6$	:	$shape(lv1_a, notch)$	:	$[\,]$
$s3_1$	:	$be\_end(lv2_a, lv2)$	:	$[\,]$
$s3_2$	:	$part\_of(lv2_b, lv2)$	:	$[\,]$
$s3_3$	:	$be\_end(lv2_c, lv2)$	:	$[\,]$
$s3_4$	:	$phase(lv2, solid)$	:	$[\,]$
$s3_5$	:	$shape(lv2, beam)$	:	$[\,]$
$s3_6$	:	$shape(lv2_a, notch)$	:	$[\,]$
$s4_1$	:	$part\_of(spbd_a, spbd)$	:	$[\,]$
$s4_2$	:	$part\_of(spbd_b, spbd)$	:	$[\,]$
$s4_3$	:	$part\_of(spbd_c, spbd)$	:	$[\,]$
$s4_4$	:	$phase(spbd, solid)$	:	$[\,]$
$s4_5$	:	$posture(spbd, level)$	:	$[\,]$

$s5_1$	:	$part\_of(spl_a, spl)$	:	$[\,]$
$s5_2$	:	$part\_of(spl_b, spl)$	:	$[\,]$
$s5_3$	:	$construction(spl, spiral)$	:	$[\,]$
$s5_4$	:	$material(spl, steel)$	:	$[\,]$
$s6_1$	:	$part\_of(crk_a, crk)$	:	$[\,]$
$s6_2$	:	$part\_of(crk_b, crk)$	:	$[\,]$
$s6_3$	:	$part\_of(crk_c, crk)$	:	$[\,]$
$s6_4$	:	$shape(crk, beam)$	:	$[\,]$
$s6_5$	:	$mid(crk_a, beam)$	:	$[\,]$
$s6_6$	:	$shape(crk_a, hole)$	:	$[\,]$
$r1$	:	$on(obj, o1)$	:	$[\,]$
$r2$	:	$fix(o1_a, lv1_b)$	:	$[\,]$
$r3$	:	$fix(o1_b, lv2_b)$	:	$[\,]$
$r4$	:	$on(lv1_a, body)$	:	$[\,]$
$r5$	:	$on(lv2_a, body)$	:	$[\,]$
$r6$	:	$on(lv2_c, lv1_d)$	:	$[\,]$
$r7$	:	$on(lv1_c, spbd_a)$	:	$[\,]$
$r8$	:	$fix(spbd_b, spl_a)$	:	$[\,]$
$r9$	:	$fix(spl_b, body)$	:	$[\,]$
$r10$	:	$on(crk_b, spbd_c)$	:	$[\,]$
$r11$	:	$throgh(shaft, crk_a)$	:	$[\,]$

(a)

ra	:	$connect(o1, spl_a)$	:	$sl([\,],[r2, r3, r4, r5, r6, r7, r8])$ $J20$
rb	:	$on(spl_a, crk_a)$	:	$sl([\,],[r8, r10])$ $J21$

$F1$	::	$sl([s1_3, r1],[\,])$	$J11$					
$F2$	::	$sl([L5, L6, \cdots, L20],[\,])$	$J12$					
or	:	$sl([L21, L14, L15, L16, L22, L20],[\,])$	$J12'$	$L5$	::	$sl([s1_3],[\,])$		$J25$
$F3$	::	$sl([\,],[\,])$	$J13$	$L6$	::	$sl([r3],[\,])$		$J26$
$F4$	::	$sl([s1_1, s2_1, s2_2, s2_4, r2, r4],[\,])$	$J14$	$L7$	::	$sl([s3_2, s3_1, r5, s3_3, s3_5],[\,])$		$J27$
$F5$	::	$sl([s1_2, s3_1, s3_2, s3_4, r3, r5],[\,])$	$J15$	$L8$	::	$sl([s3_3, s3_5, r6, s2_3, s2_4],[\,])$		$J28$
$F6$	::	$sl([s2_3, s2_5, s2_6],[\,])$	$J16$	$L9$	::	$sl([s2_2, s2_1, r4, s2_3, s2_5],[\,])$		$J29$
$F7$	::	$sl([F4],[F6])$	$J17$	$L10$	::			$J30$
$F8$	::	$sl([F4, F6],[\,])$	$J18$	:		略		:
$F6'$	::	$sl([s3_3, s3_5, s3_6],[\,])$	$J16'$	:				:
$F7'$	::	$sl([F5],[F6'])$	$J17'$	$L20$	::			$J40$
$F8'$	::	$sl([F5, F6'],[\,])$	$J18'$	$L21$	::	$sl([ra],[\,])$		$J41$
$F9$	::	$sl([ra],[\,])$	$J19$	$L22$	::	$sl([rb],[\,])$		$J42$
or	:	$sl([s2'_2, s4_1, s4_4, s4_5, r6, r7, r8],[\,])$	$J19'$					

$\overline{G1}$	:	$Contradiction$	:	$sl([\,],[G1])$	$J1$
$G1$	:	体重を測定	:	$sl([F1, F2, F3][\,])$	$J1'$
$\overline{G2}$	:	$Contradiction$	:	$sl([\,],[G2])$	$J2$
$G2$	:	踏台は水平	:	$sl([F4, F5, F4', F5', s1_4],[\,])$	$J2'$
$\overline{G3}$	:	$Contradiction$	:	$sl([\,],[G4])$	$J3$
$G3$	:	体重の偏り...	:	$sl([F9],[\,])$	$J3'$

(b)

図 5.6: ノードリスト

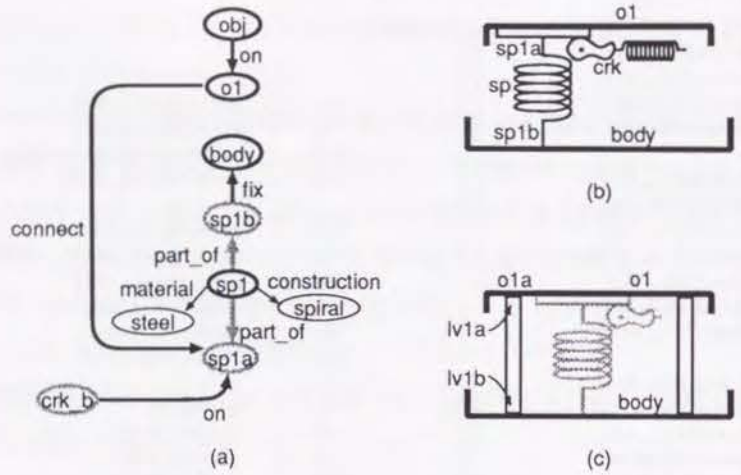


図 5.7: 体重計の最小構造 (a)(b) と基本機能 (G2) を回復した構造 (c)

なお, L5 から L20 のそれぞれを成立させる構造条件は, モード IV(図 4.7参照) で抽出され, それらを図 5.6(b) に示す Justification ($J25, J26, \cdots, J40$) で表す。

(1-b) : 物理因果連鎖の統合

物理因果連鎖の結合則 (図 5.4) を適用し, 目標とする機能を達成するための最小構造を生成する。その結果, 図 5.5(a) に示す因果連鎖は図 5.5(b) となる。上記の (1-a) の過程で得られた構造のうち $L5$〜$L13$ および $L17$〜$L19$ は, 物理量の伝達経路として冗長であり, それぞれ 1 つの因果関係 $L21, L22$ に統合される。これは, 踏台を直接バネの一端およびクランクの一部に接続することによって構造の最小化が可能であることを示している。

(1-c) : 構造の最小化

上記の物理因果連鎖の統合の結果, $F2$ を達成するには, 物理因果連鎖 $L21$, $L14$, $L15$, $L16$, $L22$, $L20$ が成立すればよいことが, さらにつぎに示す $L21$ および $L22$ を成立させる構造条件が導き出される。

$$ra : connect(o1, spl_a)$$
$$rb : on(spl_a, crk_a)$$

最小構造では, 支持バネの一端 (spl_a) に直接クランク (crk_a) と踏台 ($o1$) を接続されている。意味ネットワーク表現された最小構造, およびその外観を図 5.7(a)(b) に示す。TMS には, ここで新たに生成された構造および物理因果関係 $ra, rb, L21, L22$ が, つぎに示す正当化と共に生成される。さらに, ノード $F2$ に新たな正当化 $J12'$ が与えられる。

ra	:	$sl([\,],[r2, r3, r4, r5, r6, r7, r8])$	$J20$
rb	:	$sl([\,],[r8, r10])$	$J21$
$L21$	:	$sl([ra],[\,])$	$J41$
$L22$	:	$sl([rb],[\,])$	$J42$
$F2$	:	$sl([L21, L14, L15, L16, L22, L20],[\,])$	$J12'$

ここで, $L21$ を正当化する ra に対する justification は, $L5$ から $L13$ を Out にするための構造素間関係が含まれている[2]。これにより, 基本機能を満たす最小構造ができる。このとき In 状態にあるのは $F1$〜$F3$ とそれぞれの foundations である。元の構造でなく最小化構造で $G1$ を達成することは, TMS では $F2$ の正当化 ($J12, J12'$) の内, $J12'$を選ぶことで実現する。これにより, 依存関係をたどって ra, rb が In となり $r2$〜$r8$, $r10$(元の構造の一部) が Out 状態になる。

(2-a) : 他の機能との干渉関係の発見

体重計には, 「体重を測定する ($G1$)」他に, 「踏台は水平である ($G2$)」ことが要求される。$G2$ を満たす構造として, 踏台に適当な面積をもたせ ($s1_4$), その四隅を剛体で支持する方法が考えられる。これを以下に示す新たなノードで TMS に伝える。また, $G1$ と同様に $G2$ は必ず満たされなくてはならない基本機能なので, 同時に$\overline{G2}$を TMS に与えておく。

$\overline{G2}$	$Contradiction$	:	$sl([\,],[G2])$	$J2$
$G2$	踏台は水平	:	$sl([F4, F5, F4', F5', s1_4],[\,])$	$J2'$

ここに, $F4, F4', F5, F5'$それぞれは踏台を一点づつ, 計 4 点で支える機能を表す。たとえば $F4$ を満たすには, 踏台の一点 ($s1_1$) を剛体 ($s2_4$) の一端 ($s2_1$) によって支持し ($r2$), 別の一端 ($s2_2$) で本体と接続する ($r4$) 必要があり, これを正当化 ($J14$) で表す。$F5$ も同様であり $J15$ で表す。$F4', F5'$は, $F4, F5$ と対象の位置関係にある同じ構造によって成立する機能であるため, ここでは省略する。

$F4$	:	$sl([s1_1, s2_1, s2_2, s2_4, r2, r4],[\,])$	$J14$
$F5$	:	$sl([s1_2, s3_1, s3_2, s3_4, r3, r5],[\,])$	$J15$

また, 元の構造に存在したテコの構造との違いはつぎのように表す。

[2]因果連鎖を不成立にするには, 構造属性変更, および構造素間関係の変更が考えられるが, 少なくとも構造素間関係 (r) を崩すことで因果連鎖は破綻する。

$F6$: $sl([s2_3, s2_5, s2_6], [\,])$ $J16$
$F7$: $sl([F4], [F6])$ $J17$
$F8$: $sl([F4, F6], [\,])$ $J18$

$F6$ は剛体の形状 $(s2_5, s2_6)$ と両端以外のある一点が存在すること $(s2_3)$ を表している。よって, $F7$ は踏台と本体を固定する構造になり, $F8$ はテコの構造になる。$F5$ に対するノード $F6', F7', F8'$, ならびにそれぞれに対する正当化 $(J16', J17', J18')$ も $F4$ と同様に与えられる。ここにおける構造の外観を図 5.7(c) に示す。

いま, (1-a) によって最小化された構造を考えると, 「体重を計る $(G1)$」という基本機能を満たすために IN状態にあるノードは, $F1$,$F2$,$F3$ とその foundations だけなので $F4$,$F5$,$F4'$,$F5'$は, OUT状態である。よって, $J2'$より $G2$ が OUTとなり $J2$ より$\overline{G2}$が INとなるので, 2 つの機能間での矛盾の発生が発見される。すなわち, 基本機能 $(G1)$ を満たすための最小化された構造では, 第二の機能を実現できないことがわかる。

(2-b) : 干渉の回避

(2-a) によって発見された矛盾を回避するために, 新たな構造の付加を行なう。ここでは, 矛盾の発生の原因となった $G2$ を In とするために $J2'$より $F4$,$F5$,$F4'$,$F5'$,$S1_4$を In とする。すなわち, 最小化された構造に「踏台を 4 点で支持する」構造を付加する。これによ り, $G2$ は INとなる。しかし $J14$, $J15$ より $F4$,$F5$ を INとするとき, その foundations である $r2$〜$r5$ も INとなる。このとき, $r2$〜$r5(IN)$→ $ra(OUT)$→$L21(OUT)$→$F2(OUT)$ →$G1(OUT)$→$\overline{G1}(IN)$ となり, 再び矛盾の発生が発見される。よって, 矛盾の発生の原因となったの最小化構造の正当化 $J12'$を $J12$ に変更することにより矛盾の解消を図る。すなわち, 元の構造に戻すことにより機能の充足化を図る。

(2-a') : 非干渉の確認

また別の基本機能として踏台に乗る測定物の体重の偏りによって測定値が変化しないように「重さを一点で荷重する $(G3)$」ことが要求される。これを TMS につぎのようなノードで与える。

$\overline{G3}$: $Contradiction$: $sl([\,], [G3])$ $J3$
$G3$: 重さを一点で荷重 : $sl([F9], [\,])$ $J3'$

$F9$: : $sl([ra], [\,])$ $J19$
or : $sl([s2'_2, s4_1, s4_4, s4_5, r6, r7, r8], [\,])$ $J19'$

これらのノードは, $G3$ のためには支持バネで直接踏台を支持する (ra), または, テコ

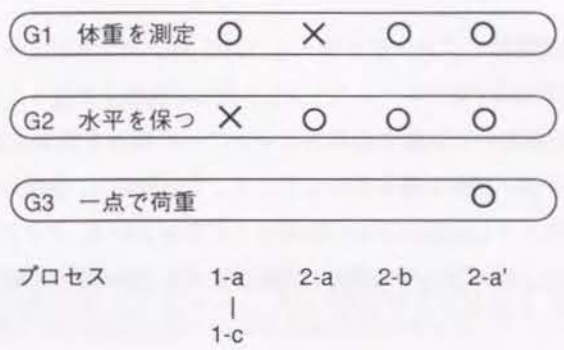


図 5.8: 体重計の深い説明

$(F8, r9)$ にかかる力を支持台に集め $(s2_2, r6, r7)$, バネは支持台に接続されている $(F8)$ 必要があることを表す。

この新たなノードが与えられることにより TMS は$\overline{G3}$を Out にするようにそれぞれのノードの状態を再び定める。この場合 $F9$ を In にすることにより他のノードに矛盾を生じさせることなくノード間の関係の整合性を保つことができる。

このような操作を繰り返して体重計の機能と構造の対応づけをとることができる。

5.4 知識抽出 (コンパイル) としての意味づけ

体重計における機能・構造の非単調な変更をまとめると, 図 5.8のようになる。ここに, ○, ×は深い説明の生成過程において得られた構造が各要求機能を満たしているか否かを示すものである。これは「体重を測定する (G1)」と「踏台は水平である (G2)」との干渉に注目しているが, これとは別に G2 と G3 に注目しても (2-a) で同様に機能間干渉を発見することができる。G2 を満たすためには, 踏台の四隅を剛体で支持する構造 (J2) が考えられる。また, 踏台に乗る測定物の偏りによって測定値が変化しないための「重さを一点に荷重する (G3)」という基本機能を満たすためには, 踏台の一点を支持するという構造 (F9) が最も単純なものとして考えられる。この二つの構造は明らかに矛盾していることがわかる。

これらの説明は, 複数の機能 (GC's) の間にどのようなかたちで干渉が生じ, それらがどのような方法で解消し得るかという, 方略的あるいはメタプラン的な知識を獲得するための手段を提供するものである。すなわち, 設計物に普遍的に内在する, 公理 1 と公理 2 により規定される設計合理性を保証するような設計知識, すなわち公理 1 および 2 をコンパイルし, これらを具体的に活用する方法についての知識が抽出されたことになる。

5.5 結論

本章では, 公理論的設計アプローチを導入し, EBG のシステムによって得られた設計事例の一般化的な構造記述 (意味ネットワーク) に対し, 公理 2 に従った構造単純化操作と, 公理 1 に従った機能間の干渉除去の操作という二つの操作を交互に施すことにより, 設計事例に対するより深い理解を得る手法を示した。この結果は, 複数の機能間の干渉が設計事例においてどのような構造によって回避されてるかという, メタプラン的な知識として抽出・獲得される。このような知識は, 次章で述べる情報伝達モデル, すなわち設計過程を情報伝達過程として, また設計物を通信路として捉えた観点から見れば, 機能ー構造連関知識 (モジュール化されたユニットコード) として解釈される。情報伝達モデルにおいては, 公理 1 は機能空間から構造空間への「意思」の疎通が誤りなく行なわれることを, 公理 2 はそれが最小の情報量によって最も効率よくなされることを要請するものである。したがって, 本章で提案する手法は, 設計の効率を向上するという意味に加えて, 公理論によって合理性・最適性が保証されているという意味でも, 非常に価値の高い知識を獲得する枠組であると考えられる。

第 6 章

抽出・獲得された知識の設計支援への利用

6.1 緒言

本章では, 4 章ならびに 5 章で提案した手法によって抽出・獲得した設計知識を用いて, 設計支援を行なう方法について検討する。理想的な支援形態は, 要求機能を満たす設計解を自動的に生成することである。しかし, 設計の下流工程においては設計パラメータの最適化をおこなう種々の方法が提案され, 計算機による支援が行われているが, より大きな効果が期待される概念レベルの改良設計については, システマティックな方法は知られていない。そこで, 1 章にも述べたように近年の設計支援研究はつぎに示す二つの方向で進められている。

第一には, 限定した判断基準のもとで計算機により自動設計を行なわせ, 提示された設計解の総合的な判断は設計者に委ねるという方法である。設計活動の中から数値処理可能な部分をモデル化し最適化手法を適用する方法, 制約充足問題としてモデル化することができる部分に対して充足アルゴリズムを適用する方法など, 従来から提案されている設計プロセスモデルはこの方法に分類することができる (たとえば [23], [28], [29], [31])。第二には, 近年注目されている研究にみられるように (たとえば, [38], [39]), あくまで設計者自体に主体性を置き, 動的に移り変わる設計フェイズを判断し, 最も適した情報を提供するシステムとして計算機を利用する方法である。

6.2節では, 前者の方法として, 遺伝的アルゴリズム (Genetic Algorithm: GA) を利用した設計解候補生成システムについて検討する [73]。このシステムは, 4 章で提案した手法で獲得された知識を統合して, 目標機能を達成する物理因果連鎖, および構造を導出するものである。導出された結果は, 過去の設計例を参考にして, それらの機能の組合せを変えることによって得られた, 意外性のある新しい概念レベルでの設計解となる。

6.3節では, 上述の後者の支援方法として, 4 章および 5 章で提案した手法により獲得

された設計知識を, 設計者が効果的に利用することができるハイパーテキスト環境のありかたを検討する。まず, 設計活動を**情報伝達過程**としてモデル化する。このモデルでは, 設計者の意図が, 機能空間上で種々のコード (符号) を介して物理因果連鎖にコーディング (符号化) されたのち, 構造空間上でのコードを用いて組織化された構造体へデコード (復号化) される。このとき設計公理は, 機能空間上での正しいコーディングおよび構造空間上での効率的経済的なデコーディングを要請するものである。

このような見地から, 4 章で示した手法は情報伝達過程におけるマクロコード, 5 章で示した手法はモジュール化された機能 (群)-属性 (群) 連関知識, すなわち情報伝達過程におけるユニットコードを獲得する手法として位置付けることができる。

これらの断片的なコードが情報伝達モデル上で連関し, 要求された機能から設計解に至る情報伝達経路が形成される。その結果は, 全体としての設計概念となる。この際, 設計者が参照する設計情報は種々の形態を持ち, かつ互いに複雑に関連したものとなる。これらを統一的な形式で表現し, 時期を得た設計情報の参照を可能にする形で設計活動を支援するために, ハイパーテキスト環境を導入する。

6.2 設計解候補生成による支援

6.2.1 生成–検査サイクルおよび事例組合せによる設計解候補生成

概念設計段階において人間が新規設計をする際, 新規な概念を考え出すことは少なく, 何らかの形で過去の設計物を参照したり, あらかじめ分かっている機能を組み合わせたりするものである。同様に, 計算機による概念設計プロセスを, 全く新しい概念を無の状態から創出するのではなく, 過去の多くの設計物を参考にし, そこに用いられている知識の組合せを変えることによって, 意外性をもった新しいシステムを生成するプロセスとして捉えることができる。このような設計プロセスモデルは, 設計問題を対象とした知識工学の分野でよく見られる。たとえば, Ulrich らによって提案された概念設計プロセスモデル [33] はファスナーの新規設計に応用されている。しかし, 設計対象記述言語がファスナーにしか利用できないこと, また表層的な構造を記述するものであり, 異なったラベルが与えられた構造が実は同じ機能を達成する構造属性をもつことが表現できないという問題がある。さらに, 新たな設計解の提示にとどまり, その評価が行なわれていないことも問題である。

このような問題を解消するために, 概念設計解候補生成システムのもつ能力として,

1. [知識ベース構築] 過去の設計物を解析し, 表層的な構造だけでなく, 機能およびそれらの間の関係の合理性の裏付けとなる基本原理 (機械設計物の場合には物理因果法則) に関する知識を, 解析対象事例に依存しない一般的な形式で抽出し, 知識ベースに蓄えること

2. [統合による解候補生成] 知識ベースにある知識を再結合し, 要求された機能を達成することのできる構造 (設計解) の候補を生成すること

3. [評価] 設計解候補を評価すること

の 3 点が必要となる。

知識ベース構築 知識ベース構築に関しては, 4 章において, 価値工学における機能分析の立場から, 物理因果連鎖の成り立ちに注目した構造物理解のための階層的設計知識を用いて, EBG 手法に基づいて知識獲得を行なう方法を提案した。このシステムは, 機能を物理因果連鎖に展開するための知識 (以下, **複合設計概念**あるいは**マクロコード**と呼ぶ), ならびに物理因果関係を成立させるための条件としての構造属性知識を生成する。

統合による解候補生成 設計解候補の生成は, 目標とする機能を物理因果連鎖に展開し, それを実現する構造属性集合を探索するプロセスとして, 以下のように定式化することができる。

1. 与えられた設計要求に対応する目標を目標空間 (G) から探しだす。

2. その目標を達成する物理因果連鎖を, 機能を物理因果連鎖に展開するマクロコードを組み合わせて生成する。

3. 構造属性空間 (S) の中から, 連鎖を構成するユニットである物理因果関係の達成に必要な構造属性を調べる。

4. 上記の構造属性をもつ実体を探索する。

上記のプロセスでは, 事例に依存したマクロコードを用いた場合, 過去の設計物を再現するのみに留まる可能性がある。すなわち, 創造的設計のためには, ある程度一般化されたマクロコードを用い, かつそれらのかつてない組合せを考える必要がある。

問題点 2 に関しては, 知識ベース内を盲目的に探索する方法では, 広大な探索空間のすべてを調べることになってしまい, 計算時間の爆発が予想され, 現実的でない。これに対し, 全ての知識間に利用すべき優先順序をあらかじめ明示的に与えるなどの手法が考えられるが, 意外性をもった新しい設計解の候補を提示するという目的にはなじまないと考えられる。

このように, 設計問題を組合せ問題として定式化すると, 組合せ的爆発などの問題への対応が必要となる。

構造物の評価 どのような尺度で構造物間に優劣をつければよいかを考察する必要がある。本研究では, 構造物中のエネルギの流れを物理因果連鎖として捉え, この連鎖のつながり具合を調べることによって評価する方法, ならびに Suh の設計公理に基づいた方法を提案する。

6.2.2 GA を用いた設計解候補生成プロセス

概念設計解候補生成を, マクロコードの統合による設計解候補生成およびその評価という組合せ最適化問題として定式化した場合, 以下の点を考慮しなければならない。

1. 組合せ的爆発を回避する。
2. 総合的判断をする設計者に対して, 特定の側面から評価を行なった解候補を参考例として提示する計算機システムという立場から, 唯一最適解よりも準最適な満足解を複数提示することが望ましい。
3. 概念設計解を評価する形式化された手法はほとんど知られていない。したがって, 定量的な評価値を与え, 連続性, 微分可能性をもつ目的関数の定義は困難である。

この種の問題は最適化が困難であり, 様々なヒューリスティクスを用いて試行錯誤的に行なわれてきたが, 近年は比較的容易に準最適な解を探索する組合せ最適化アルゴリズムの発展とともに, 自動化が試みられている。しかしその多くは, 与えられた制約の下で目的関数を最大化するような変数値の組を決定するパラメトリック設計を対象としている。これらは, 本研究の定式化のような問題点 1 から 3 の全てに対する考察が必要な問題を対象としたものではない。そこで, 本節では, 2.4.2節で述べた遺伝的アルゴリズム (Genetic Algorithm: GA) を導入した設計解候補生成プロセスについて検討する。

近年, 生物の原理を形式的に実現し, 工学的に応用する技術として, 人工生命 (Artificial Life: AL) が注目を集めている [52]。このような枠組の一つに位置づけられる GA は, 生物進化の機構を形式化したものであり, 最適化問題や探索問題などに有効な手法である。GA が従来の最適化手法に優れている点は, 問題の性質に大きく依存しないことや, 最適化の目的関数に連続性や微分可能性などの制約が必要ないこと, および定義域がどのような集合であっても構わないことである。また, GA は仮説生成, 評価, 選択の繰り返し過程であり, 設計型問題への基本的アプローチである生成-検査法を実現するものと捉えられる。この GA を設計問題の領域に適用することができれば, つぎの効果が期待される。

- 知識の優先順位や過去の知識の組合せパターンに左右されることなく, ランダムに設計解を生成するため, 過去にはない知識の組合せから導かれる, 意外かつ優れた解候補の生成が可能である

- 唯一無二の設計解ではなく, 複数の解候補を生成することができる
- 広大な探索空間を全て調べる必要がなく, 計算時間の爆発を抑えられる

本研究では, 4 章で提案した手法で得られた一般化機能系統図を遺伝子として与え, GA における「表現型」を物理因果連鎖とする。物理因果連鎖を構成するユニットである物理因果関係には, その関係が成立するための構造的条件が定められている。したがって, この表現型は, 設計物における「(要求機能を達成するための) 最小限の構造属性」を表現することになる。この場合, 機能系統図が交叉 (クロスオーバ), 突然変異 (ミューテーション) を行なうことになるが, これらは概念設計に置き換えて考えると, 「機能の組変え」に相当し, 新たな機能の組合せを探る「発見的創造」手法を実現するものである。

遺伝子の表現法

GA における遺伝子は, 一般に一次元の数字や, 記号の列を用いて表現される。ところが, 本研究が対象とするのは木構造をもつ機能系統図である。この木構造を, 今までの GA のような一次元の配列に変換するのは不自然であり, クロスオーバーやミューテーションの定義が困難になる。そこで, 木構造 (機能系統図) そのままを遺伝子として採用し, この上で上記の操作を行なう。

一般に, 遺伝子の表現法を決定するための評価規範として, **完備性 (completeness)**, **健全性 (soundness)**, **非冗長性 (non-redundancy)** がある [74] [75]。以下に, それぞれの規範について検討する。

・**完備性 – 問題空間上のすべての点 (解候補=表現型) は GA 空間上の点 (遺伝子) として表現されることが望ましい：** 無作為に並べた物理因果連鎖 (解候補) は, 対応する機能系統図をもたない可能性がある。しかし, 用いる知識が完全であるとするならば機能系統図によって説明できないような物理因果連鎖は存在しないといえる。よって, 解空間の各要素は遺伝子空間の要素と必ず対応づけることができる。すなわち, 完備性は満たされている。

・**健全性 – GA 空間上のすべての点 (遺伝子) は問題空間上の点 (解候補) に対応づけられることが望ましい：** すべての機能系統図 (遺伝子) には物理因果連鎖 (解候補) が存在しており, 健全性は満たされている。

・**非冗長性 – 問題空間上の点 (解候補) と GA 空間上の点 (遺伝子) は 1:1 に対応づけられることが望ましい：** ある物理法則列 (解候補) は, 複数の機能を果たす可能性があるので, 非冗長性は一般に成立しない。

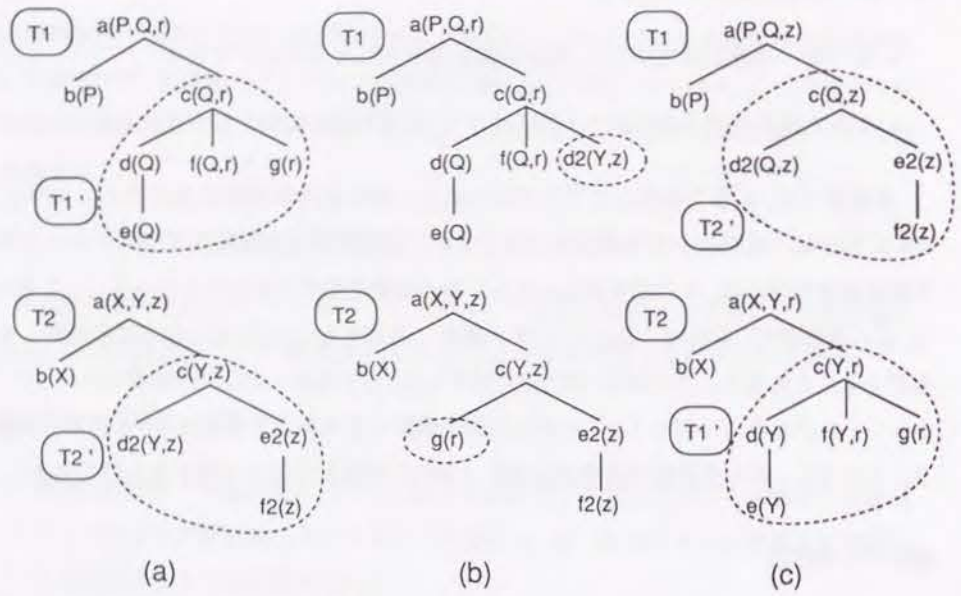


図 6.1: 木構造の組変え

遺伝子組変えの方法

遺伝子とする機能系統図は木構造である。GA を木構造などのグラフ構造体へ拡張する試みは, "構造的表現の適応学習 [76]" として研究され, その有効性も示されている。その枠組は適応的な学習のための枠組として提案されてはいるが, 本研究で用いる手法もそこで用いられている組変えの手法と基本的には同じものである。

すなわち, 組変えは, 部分木を取り替えることによって行なう。まず, 図6.1(a) に示す 2 つのリスト (木) T_1, T_2を考える。このリストに対してクロスオーバーを起こす場合, 組変え可能な部分は a, b, c の 3 ヶ所である。たとえば, 図6.1(a) の場合 $c \Leftarrow (d \wedge f \wedge g) \vee (d2 \wedge e2)$ という知識を用いている。ここで仮に, g と d2 を組変えると, 図 6.1(b) に示す結果となり, $c \Leftarrow (d \wedge f \wedge d2) \vee (g \wedge e2)$ という存在しない知識を用いることになる。つまり, 機能系統図の正当性が失われ, 全く意味の無い構造を作ることになる。

いま, 仮に c に組変えが起こった場合を考える。このとき, ともに c を根とする部分木 (図6.1(a) における T_1', T_2'), を入れ替える。ただし, 単に入れ換えるだけではなく, 変数の拘束関係をも考慮しなければならない。上記の T_1'と T_2' をそのまま入れ換えると, お互いの変数が違ったままで, 最初に親の木 T_1, T_2 がもっていた情報を失ってしまうことになる。このリストで変数の拘束関係を考えると, 図6.1(c) に示す 2 つの遺伝子が得られる。

突然変異

つぎに, 突然変異を定義する。これについても, ある一箇所をランダムな数値や文字で置き換えるという一般的な方法を採用すると, 整合性の無い機能系統図を生成することになる。たとえば, T1 の g(r) を h(T) で置き換えると, 図 6.2に示すような木 T_3 が生成され, $c \Leftarrow (d \wedge f \wedge g) \vee (d2 \wedge e2)$ に反することになる。そこで, 突然変異を, 全く初めからランダムにマクロコードを組み合わせた木を生成することで実現する。

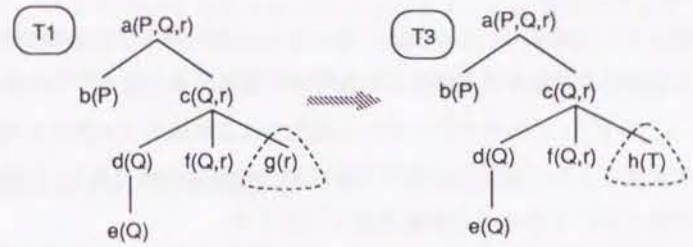


図 6.2: 突然変異によって発生した致死遺伝子

ここで, GA を導入した設計過程のモデルについてここまで述べてきたことを整理するために, 上記のアルゴリズムを導入した GA の過程を以下に示す。

1. 初期の遺伝子プールを作るために, 機能系統図を必要な数だけ作る。
   (a) 階層状に構造化された設計知識空間 (およびマクロコード), G, P, F, L の順に, 要求された目標機能の十分条件を探索する。
   (b) 要求を満たす知識が複数存在する場合, ランダムに一つを選ぶ。
   (c) 機能系統図は, 一般に, 物理因果関係 (およびそれに付随した構造属性) を末葉としている。したがって, ここでの十分条件の探索は物理因果知識空間 (L) で終端することになる。
2. 遺伝子プールの遺伝子を評価する。
3. 評価値を基にクロスオーバー, ミューテーションを行ない, 新たな世代の遺伝子プールを作成する。
4. 評価値が収束するとき, これを解候補として出力する。そうでない場合, 2 に戻る。

6.2.3 設計物の評価

設計物を評価する方法に関する研究は, 設計診断の問題としてとくに定性推論に関する領域で盛んである。たとえば, 中島らは構造物の定性的な表現を利用して, 機械の設計

診断と設計修正案の提示を行なうシステムを実現している [77]。また, 診断対象を表現するのに定性物理を導入し, 定性物理の枠組の挙動推論を利用して診断を行なう方法などが提案されている [78]。しかし, 実際に開発されてきたこの種のシステムに対しては,

- 評価基準が個々の設計において異なるので, システム自身に汎用性が無い
- 同様に, 使用する知識が対象領域に偏りがちで, 汎用性が無い

などの問題点がある。

これらを解決するためには, 広く設計に用いることのできる評価基準が必要である。そこで, 物理的設計物を規定する原理的な知識である物理因果法則の列を対象とした評価について考える。つまり, このモデルにおける表現型は目標機能から展開された物理因果連鎖と, それが成立するのに最低限必要な構造空間への要求の組であり, この表現型に対して, 本研究ではつぎに示す 3 つの評価基準を提案する。

1. その設計物の (目標達成の) 背後にある物理因果連鎖上でのエネルギの流れを追跡し, その矛盾や無駄に基づいた評価
2. 物理因果連鎖そのものに対しての設計公理に基づいた評価
3. 構造に要求される最低限の属性に対しての設計公理に基づいた評価

これらの方法は, 物理因果連鎖, すなわち物理的設計物を表現する普遍的な「深い知識」に対する評価を行なうものである。

評価の対象

「目標機能から展開された物理因果連鎖およびそれを成立させるために必要となる構造属性」を評価の対象とする。これは 3 章で述べた設計物の階層構造モデルに基づくものである。この設計案生成の過程を輸送機器の設計を例として検討し, 評価対象となる表現について考察する。

まず輸送機を設計したいという要求を, つぎのように記述する。

build(transporter(A, B)).
「A を輸送する B を設計せよ」

つぎに, これを達成することのできる機能を知識ベースから探し出す。知識ベースには 4 章で提案した手法によって獲得された情報伝達プロセスのマクロコード, すなわち, ある機能とそれを達成する手段を複数個収めておく。

dt(transporter(Load,Machine),(removal(Load,Machine),mobile(Machine))).
「Load(荷) を輸送する Machine は, Load を Machine に装着 (removal) でき, Machine が移動するもの (mobile) であればよい」

このような知識に従って機能を展開し続ければ, 最終的には物理因果関係の連鎖を得ることができる。個々の物理因果関係は,

p_ p(X,torque,Y,torque)
「X にかかるトルクと Y にかかるトルクには関係がある」

という形をしており, これらがいくつか集まって一つの因果連鎖を形成する。

物理因果連鎖に対するエネルギ追跡法

物理因果連鎖は, エネルギや情報の流れ[1]であると捉えることができる。もしすべての設計物がこの物理因果法則の連鎖で表されるとするならば, すべての設計物にはエネルギの流れがあることになる。つまり, エネルギの流れを評価することができれば設計を評価する際の共通の基準とすることができる。

本研究では, 先にも述べたように, 物理法則もしくは因果法則が成り立つときのエネルギの流れの矛盾や無駄に着目するものである。

- 矛盾生起 (図 6.3)
  たとえば,

  p_ p(sun,ray,1,heat_ energy).
  「太陽の光から得られるエネルギと
  構造 1 に発生する熱エネルギには関係がある」
  p_ p(1,heat_ energy,2,explosion).
  「構造 1 に発生する熱エネルギと
  構造 2 にかかる爆発力には関係がある」

  という物理因果法則連鎖があったとする。太陽光のもつ熱エネルギは, 爆発を起こすほど高いものではないから, この法則列は矛盾を生じ, 成立しにくいものになる。よって, このような法則列に対してはマイナス点を与える。

- 無駄の存在 (図 6.4)
  つぎのような物理法則列を考える。

[1]以下では, 「エネルギおよび情報の流れ」を簡略化して「エネルギの流れ」と呼ぶ。

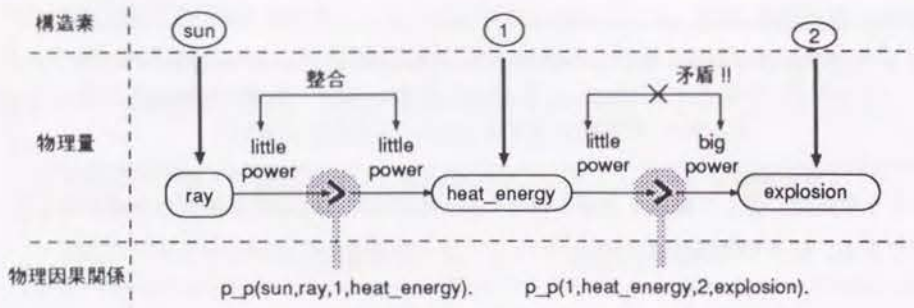


図 6.3: エネルギの流れの矛盾

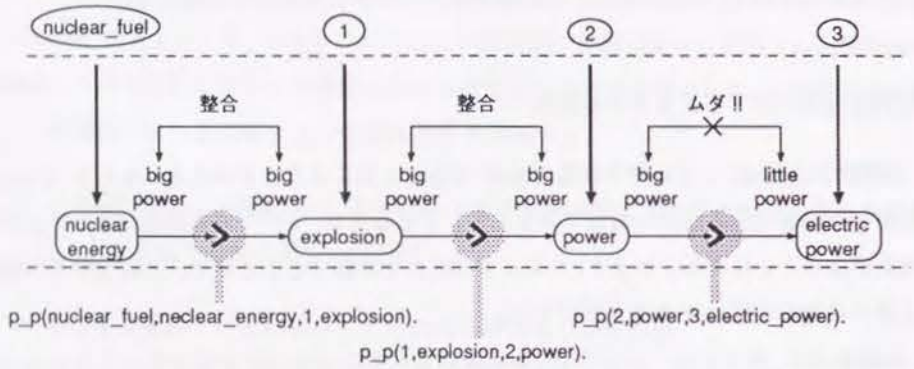


図 6.4: エネルギの流れの無駄

p_ p(nuclear_ fuel,nuclear_ energy,1,explosion).
「核燃料のもつ核エネルギと構造 1 にかかる爆発力には関係がある」
p_ p(1,explosion,2,power).
「構造 1 にかかる爆発力と構造 2 にかかる力には関係がある」
p_ p(2,power,3,electric_ energy).
「構造 2 にかかる力と構造 3 の発生する電気的エネルギには
関係がある」

核爆発は非常に強大な力を生み出す。しかし, いま, 構造 3 において必要とする電気的エネルギ量が少なくてもすむ (これは電気をどのような目的で使用するかによる) と仮定すると, このような法則列を用いた設計は大量のエネルギを無駄にすることになる。つまり, マイナス点が与えられる。

以上のようにして物理因果連鎖レベルでエネルギの流れを追跡する手法は, 一般的な設計物に対する評価を可能にする。

物理因果連鎖に対する設計公理論に基づく方法

2.3.2節で示した公理論的設計論における 2 つの公理に照らして, 設計解候補の評価法について検討する。設計公理 1 は, 機能の独立性を保たねばならないことを要請している。本研究の場合, 単一の機能だけに注目しているため, 少なくとも一つの機能が必ず達成されるという意味では公理 1 は保証されている。設計公理 2 は, 設計物に含まれる情報量は少ない方がよいことを要請するものである。すなわち, 対象とする設計物が用いている物理因果法則の数を最小化する指向性を持ち, これは, 複雑すぎる物理法則列にはマイナス点を与えることにより評価される。

たとえば, ガソリンエンジンの例として, 以下の二つを考える。

PLAN1. ガソリンエンジン
p_ p(gasoline,heat_ energy,1,explosion).
p_ p(1,explosion,2,power).
p_ p(2,power,machine,pro_ forth).

PLAN2. ガソリンエンジン
p_ p(gasoline,heat_ energy,1,explosion).
p_ p(1,explosion,2,power).
p_ p(2,power,3,electric_ power).
p_ p(3,electric_power,4,power).
p_ p(4,power,machine,pro_ forth).

これらは同じ機能を果たしているにもかかわらず, PLAN1 では 3 つ, PLAN2 では 5 つの物理因果法則を用いており, PLAN2 の方がより複雑な提案になっていることがわかる。これは, PLAN2 の方で力 (power) を推進力 (pro_ forth) に変換するときにいったん電気 (electric_ power) エネルギに変換して伝えるという余分な道のりを経由するものであり, このような無駄な道のりにたいしてマイナス点を与えればよい。本研究では, 無駄な道のりを省くために, 逆に連鎖を形成している物理因果関係の数が少ないほど高得点を与えている。

構造属性に対する設計公理論に基づく評価

先に示した二つの評価法は, 基本機能を詳細化して, 物理因果法則連鎖に展開し, 物理因果連鎖に対する評価を行うものであった。つぎに, 個々の物理因果関係を成立させるために, どのような構造上の性質が要求されるかに注目し, これに対する評価法を考える。以下, 「流速測定器」を例にとり考察する。

輸送機の例と同じように, 基本機能を物理因果連鎖に展開する。連鎖の構成要素は, たとえば,

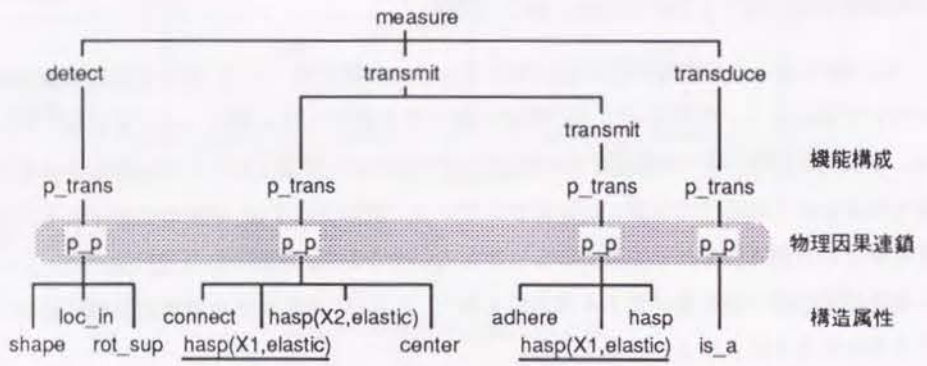


図 6.5: 構造レベルに拡張した機能系統図

p_p(X,stress,Y,stress)
X にかかる圧力と Y にかかる圧力には関係がある

といった物理因果関係であり, 個々の物理因果関係には, それが成立するための構造属性が与えられている。構造属性とは, たとえば, adhere, connect, fix などの構造素間関係, hasp(has property), is_a, phase, shape などの構造素のもつ表層的な構造情報であり, これらを書き加えると機能系統図は図 6.5に示すように, 物理因果連鎖から, 構造属性のレベルにまで拡張される。

構造を直接評価する基準としてつぎの二つを考えることができる。

1. 要求される構造属性の数
2. 構成部品の数

その理由を以下に示す。

Function Sharing Function Sharing[33] は, まずある機械が要求された通りに機能するために必要な複数の詳細な機能に注目し, それぞれの機能を達成する構造を個別に生成する。つぎに, それぞれの構造で共有 (share) できるものを探し, 設計物の構造を精錬する設計方法論である。つまり, 同じ機能を果たすなら機械的に複雑な構造よりも単純な構造の方がよいという考え方である。この考え方は, 「構成されている部品数は少ないほど高得点を与える」という評価基準と本質的に同じであると考えられる。

経済合理性 個々の設計物全体の製造コストは, 主に部品の製造コストと組み立てのコストから構成される。部品の製造コストの点でみると, 一つの部品のもつ構造属性が少ないほど, すなわち, 単純な構造ほど一般的にコストは安くなる。しかし, 逆に組み立てコストの点からみると, 一つの部品に多くの構造属性をもたせ部品の数を少なくするほど, 組み立てにかかるコストが下がる。

また, 故障という点から考えると, 部品の数が少ないほど, 故障原因も少なくなると考えられるが, 逆に一つの部品が多くの機能達成に関与しているほど, いったんその部品に故障が起きたときの被害が大きくなる。

このように, 同じ数の構造属性が要求されているならば, それを少数の部品で実現する戦略をとる場合にも, 多くの部品で実現する戦略をとる場合にも一長一短がある。したがって, どちらの戦略をとるかは, 設計者がどのような設計を意図しているかによって変化するものであり, 「部品の数」による評価より「要求される構造属性の数」による評価が本質的であるといえる。そこで, 本実験では, 「要求される構造属性の数」に注目した評価法を用いる。実際の評価では, 単純に機能系統図に含まれる構造属性の数を数えるのではなく, つぎのようなものを削除することにする。

重複する構造属性 図 6.5に示す $hasp(X1, elastic)$ のように, 異なった機能から要求された同じ構造属性

意味のない構造属性 たとえば, $adhere(A, A)$(A と A は密着している) のような自明な構造属性

6.2.4 設計解候補生成システムの実装

遺伝子生成

遺伝子は, 目標機能から領域知識 (およびマクロコード) を用いてそれを達成する十分条件を探索し, 物理因果連鎖で終端する木構造である。これに対する生成器は, 図 6.6(a) に示すように, 4 章で示した (EBG による) 機能系統図生成エンジンから一般化の機能を取り除き, 一部修正することにより実現される。

図 6.6(a) の述語 seed における第 1 引数は, 現在探索中のノードを示し, 第 2 引数が backtrack しながら全ての説明木を生成する。seed の第 1 行目は, 現在探索中の第 1 引数に対してこれを満たす物理法則の全てを調べ (findall), その中からランダムに 1 つを選ぶ (select)。第 2 行目は機能系統図における分岐点の場合である。第 3 行目は第 1 行目と同じく, 現在探索中の第 1 引数にたいしてこれを満たす領域知識 (およびマクロコード) の中からランダムに一つを選び, 再び探索を繰り返す。ここで, 領域知識には再帰的な記述が含まれているため, 述語 check において自分の親にあたるノードと同じ探索は行わないように再帰制御を行なっている。述語 seed の第 3 引数はこの制御のために, 現在までの探索木を保持する。

交叉

交叉 (Crossover) は, 部分木を交換することによって行なうため, 以下のようなアルゴリズムを用いた。

まず, 図 6.6(b) の crosser で, 交叉させる二つのリストから交叉できるノードを全て抜きだし (find), その中からランダムに 1 つを選び出す (select)。そこで決定されたノードを交叉点として実際の交叉を行なう (cross)。述語 cross の第 1, 第 2 引数は交叉する時点での親遺伝子で, 第 3, 第 4 引数が子の遺伝子である。第 5 引数は交叉点を示している。さ

```
seed(p_p(A,B,C,D),[pd(p_p(A,B,C,D))],_) :-
    copy(pd(A,B,C,D),pd(SA,SB,SC,SD)), findall(N,pd(N,p_p(SA,SB,SC,SD),_),X),
    select(X,S), pd(S,p_p(A,B,C,D),_).
seed((A,B),P,Tr) :-
    seed(A,Ap,Tr), seed(B,Bp,Tr), append(Ap,Bp,P).
seed(A,[P],Tr) :-
    check(Tr,A), copy(dt(A),dt(SA)), findall(N,dt(N,SA,_),X),select(X,S),
            dt(S,A,B), seed(B,Bp,[A|Tr]), append([dt(A)],Bp,P).
seed(A,[A],_) :- !,fail.
```

(a) 遺伝子生成

```
crosser(List1,List2) :-
    find(List1,List2,SL), select(SL,DL), cross(List1,List2,NList1,NList2,DL).
cross(L1,L2,NL1,NL2,C) :-
    c_cutter(L1,C,L1U,L1D), c_cutter(L2,C,L2U,L2D), !,cross_check(L1D,L2D),
    ins(L2D,L1U,NL1), ins(L1D,L2U,NL2).
```

(b) 交叉

```
persue(X,P,Sp,List,MList) :-
    p_p(X,P,Y,Q),pd(_,p_p(X,P,Y,Q),proportion), retract('p_p'(X,P,Y,Q)),
    counter, e_check(X,P,Sp,List,NList), persue(Y,Q,Sp,NList,MList).
persue(X,P,Sp,List,MList) :-
    p_p(X,P,Y,Q),pd(_,p_p(X,P,Y,Q),(S1,S2)), atom(Sp),S1 \== Sp,
    append(List,[pp(X,P,Sp)],LList), retract('p_p'(X,P,Y,Q)),counter,
    e_check(X,P,[c,S1],LList,NList), persue(Y,Q,S2,NList,MList).
persue(X,P,Sp,List,MList) :-
    p_p(X,P,Y,Q),pd(_,p_p(X,P,Y,Q),(S1,S2)), retract('p_p'(X,P,Y,Q)),counter,
    e_check(X,P,S1,List,NList), persue(Y,Q,S2,NList,MList).
persue(X,P,Sp,List,MList) :-
    p_p(X,P,Y,Q),pd(_,p_p(X,P,Y,Q),p_trans), retract('p_p'(X,P,Y,Q)),counter,
    e_check(X,P,p_trans,List,NList), e_check(Y,Q,p_trans,NList,MList),!.
persue(X,P, _,List,NList) :-
    p_p(X,P,Y,Q),pd(_,p_p(X,P,Y,Q),etc), retract('p_p'(X,P,Y,Q)),counter,
    e_check(X,P,nothing,List,NList),!.
```

(c) 評価

図 6.6: 解候補生成システムの 3 つの推論エンジン

らに, cross では親の遺伝子を交叉点で切りとったあと (c_ cutter), 組変え後の機能のつながり方を想定して, それが実際の知識に沿っているかどうかを調べてから (cross_ check), 実際の交叉を行う (ins)。

評価

評価は, つぎの 3 つのプロセスから成る。

1. 機能系統図から物理法則列を導く
2. 変数化されている部品に, 部品番号を割り当てる
3. エネルギの流れを追跡して, その妥当性を評価する (図 6.6(c))

1, 2 は, 3 のための前処理である。1 の処理で機能系統図から物理法則列を抜きだす。2 の処理は, 構成物に番号をつけて, エネルギの流れを明確にする。この処理を行なわず変数のままで 3 の段階に移ると, prolog の単一化 (unify) により違う部品を同一視してしまうことになる。そして, 3 の処理で実際のエネルギの流れから無駄や矛盾を発見し, 評価する。

ここで, 本研究の評価法では, エネルギを追跡する際に, 注目している物理法則が必要とするエネルギ量, 出力するエネルギ量などの情報が必要になることを考えて, 物理法則データベースの記述方法を以下のように拡張する。

pd(14,p_ p(X,nuclear_ energy,Y,explosion),(big,big)).
「核力を爆発力に変換すると, 巨大なエネルギを必要とし,
巨大なエネルギを発生する。」
pd(9,p_ p(X,power,Y,torque),proportion).
「力をトルクに変換するときの出力は, 入力するエネルギ量に比例する。」

第 1 引数の数字は, その知識の番号を表している。

このような表記法により追跡法を実現するのが, 図 6.6(c) の persuer である。第 1, 第 2 引数は現在注目している部品名とその部品のもつエネルギ (エネルギ種別) であり, 第 3 引数はそのエネルギ量を示している。第 4 引数が評価すべき物理法則列で, 第 5 引数に無駄, 矛盾のあった法則のリストを返すようになっている。

図 6.6(c) での主な処理内容は以下のようになる。まず, p_ p(X,P,Y,Q) で現在追跡している部品 (第 1,2 引数) に続く物理法則を用いているかを調べる。つぎに, pd(_,p_ p(X,P,Y,Q),Relation) でその物理法則にどのような入出力関係 (Relation) が必要かをデータベースから捜し出す。そして, 述語 e_check では入出力関係と現在注目している部品がもっているエネルギ量を比較し, エネルギ関係に矛盾や無駄があった場合に第 5 引数のリストに加え, さらに persuer でつぎのエネルギ関係を追跡する。そして, これに基づ

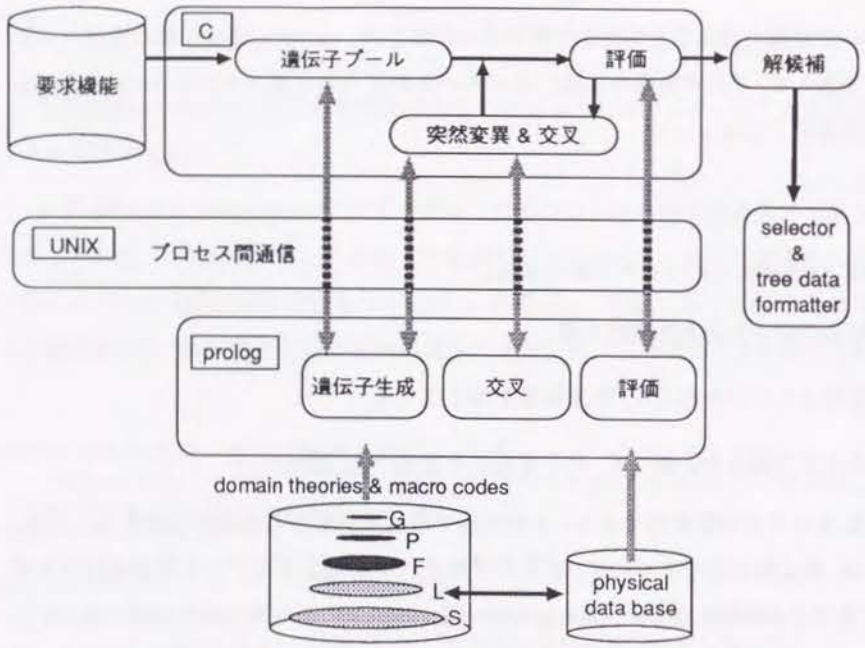


図 6.7: 解候補生成システムの概略図

き, 1 行目は比例関係の場合, 2 行目は矛盾または無駄があった場合, 3 行目は一致した場合, 4 行目はエネルギではなく位置的な移動が起こった場合, 5 行目はそれ以外の場合を処理する。

6.2.5 試作システムによる実験結果と考察

システムの概略

図 6.7に試作したシステムの構成を示す。本システムでは, ここまでに述べてきた 3 つの prolog プロセスに加えて, 実際の GA のメインルーチンの部分を C 言語で記述し, 両者の通信に UNIX プロセス間通信を用いている。また, システムの出力は C 側からリスト形式の形で出力され, 別途に用意した tree data formatter で木構造に変換することができるようになっている。

物理因果連鎖に対する評価による実験

図 6.8(a) に, 物理因果連鎖に対する評価を, 輸送器の設計を例題として行なった結果を示す。この実験では, 図 6.8(b) に示すパラメータ値を用いた。第 10 世代で, 60 個の遺伝子の内ほぼ半数の評価値が最大値 1.6に収束している。 この実験の結果の中で, 高得点を得た設計解候補の一例が図 6.9である。図 6.9(b) はシステムの出力した機能系統図であり, 図 6.9(a) は, その中に存在する物理法則列のエネルギの流れに注目し, どのような

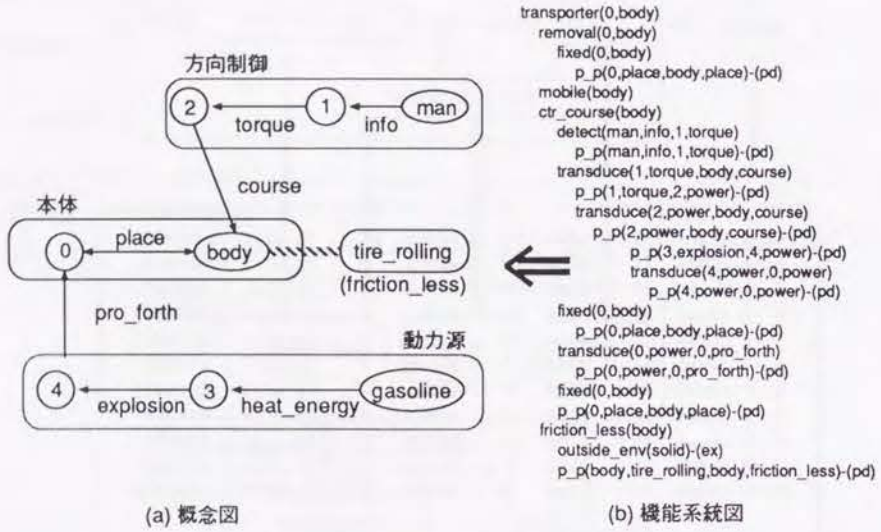

```
 0 -> count  11 .. plus  10 .. minus  51 .. (p-m)/c -4.000000
 1 -> count  11 .. plus   7 .. minus   0 .. (p-m)/c 1.571429
 2 -> count  10 .. plus   6 .. minus   0 .. (p-m)/c 1.666667
 3 -> count  10 .. plus   6 .. minus   0 .. (p-m)/c 1.666667
 4 -> count  10 .. plus   6 .. minus   0 .. (p-m)/c 1.666667
 5 -> count  11 .. plus   7 .. minus   0 .. (p-m)/c 1.571429
 6 -> count  10 .. plus   6 .. minus   0 .. (p-m)/c 1.666667
 7 -> count  10 .. plus   6 .. minus   0 .. (p-m)/c 1.666667
 8 -> count  10 .. plus  13 .. minus   0 .. (p-m)/c 0.769231
 9 -> count  11 .. plus   7 .. minus   0 .. (p-m)/c 1.571429
10 -> count  11 .. plus   9 .. minus  25 .. (p-m)/c -1.555556

                                  ≀

50 -> count  11 .. plus   7 .. minus   0 .. (p-m)/c 1.571429
51 -> count   8 .. plus   6 .. minus   0 .. (p-m)/c 1.333333
52 -> count  10 .. plus   6 .. minus   0 .. (p-m)/c 1.666667
53 -> count  11 .. plus   7 .. minus   0 .. (p-m)/c 1.571429
54 -> count  10 .. plus   6 .. minus   0 .. (p-m)/c 1.666667
55 -> count  11 .. plus   7 .. minus   0 .. (p-m)/c 1.571429
56 -> count  11 .. plus   7 .. minus   0 .. (p-m)/c 1.571429
57 -> count  11 .. plus   7 .. minus   0 .. (p-m)/c 1.571429
58 -> count  11 .. plus   7 .. minus   0 .. (p-m)/c 1.571429
59 -> count  10 .. plus   6 .. minus   0 .. (p-m)/c 1.666667

GENERATION  10
Max Vol  1.666667
Min Vol  -4.000000
Avg Vol  1.379752

User  time: 132 seconds
```

(a) 実験結果

遺伝子の数	60 個
世代数	10 世代
交叉する確率	80%
突然変異の確率	10%

(b) パラメータ

図 6.8: 試作システムによる実験結果



(a) 概念図　　(b) 機能系統図

図 6.9: 高得点を得た解候補の概念図

プランで解 (物理法則列) を提示しているかを概念的に示したものである。

また, 比較実験として, 全ての知識の組合せを順次試行し, さらに評価を行なうプロセスを実行させたが, 実験に用いた計算機の容量の関係で[2] , 用意した全ての知識を用いた推論は試すことはできなかった。そこで, 使用することができる知識の数を大幅に減らしたうえで推論を行なわせた。使用可能な知識の数を減らすために, 輸送機器のエネルギ源を 1/2 に制限している。この場合, 実行時間は約 3 時間で, 2106 通りの設計解候補案を生成しているが, これは, 用いた知識の数 (領域知識:17, 物理法則:21) を考えると膨大な時間である。なぜなら, 実際のシステムではより多くの知識が必要なことが必至であり, 知識を増やすことによって, その組合せ, すなわち実行時間は指数関数的に増大するためである。また, すべての知識を使用したと仮定すると, 計算上は 12 時間以上かけて 10000 通り近くの解候補を生成することになる[3]。これら解候補中から優秀なものを発見するのはさらに時間のかかる作業となろう。これに対して本研究の提案する方法では, たとえば図 6.8(a) に示すように, 実行にかかった時間は 132 秒という十分現実的な時間で, "満足することができる" (得点の高い) 設計解候補を複数提案している。さらに, この方法では, 知識の数を増やした場合に増大するのは初期遺伝子プールの遺伝子を生成する時間のみであり, その後の組変えプロセスや, 評価の時間に対する影響は少ない。つまり, 実行時

[2]用いた計算機 : Sun SPARCstation IPC(Memory 24MB)
用いた言語 : SICStus-prolog ver.2.1 上でコンパイル

[3]ここで用いた輸送機の例題の場合, エネルギ源を必要とするのは主に 2ヶ所である。よって, これらに 8 つのエネルギ源を当てはめる組合せが ${}_8C_2 = 27$ 通り, 実際に実験できた組合せが ${}_4C_2 = 6$ 通りとなる。よって時間も代替解の数も比例すると仮定したとき, 単純計算で考えても計算時間が約 (3 * 27/6 =)12.5 時間, 解候補は約 (2106 * 27/6 =)9477 通りとなる。

間の増大を抑えることが期待される。

しかし, 実験により以下のような問題点も判明した。

- 評価値にあまり差が出ない
  評価値を決定する要因となる物理法則データベースに登録されるエネルギ関係の記述が単純であり, マイナスポイントになるパターンが特定のものになるために起こると推測される。つまり, より詳しい記述法を導入しなければならない。

- GA のパラメータを決定する形式的な方法がない
  これは GA そのものの問題点として取り上げられることが多いが, 遺伝子プールの遺伝子の数や, 交叉・突然変異の確率などを決定する手段がない。

構造レベルでの評価をとり入れた実験

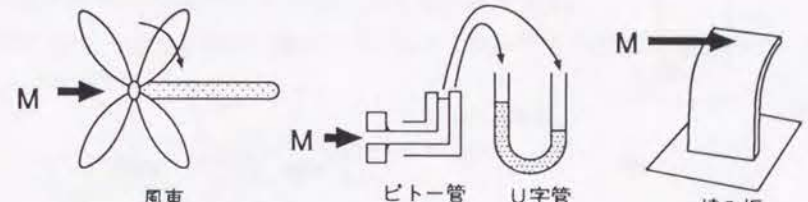


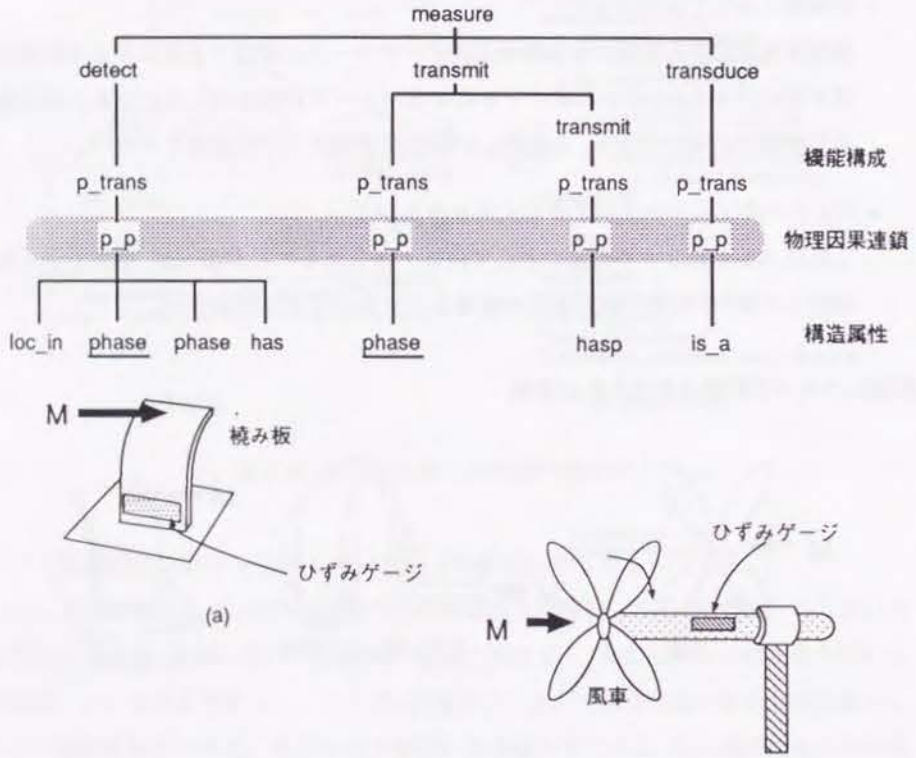
図 6.10: 設計知識を抽出した事例 (一部)

図 6.10に示す風車, ピトー管, U 字管, 桃 (たわ) み板などから知識を取り出し, その知識を再結合して流体の速度測定器を設計する実験を, 構造レベルの評価を取り入れて行なった。このときの GA のパラメータ値を以下に示す。

遺伝子の数	60 個
世代数	20
交差確率	80 %
突然変異の確率	10 %

図 6.11(a) に示すのは, 得られた設計解候補の一つである。この機能系統図における構造属性の数は, 同じもの ($phase(\_1761, solid)$, 図 6.11(a) 下線部) を除くと 6 つである。また, 比較をするために, すべての組合せを調べたところ 76 通りの設計解が得られた。この中で評価の高い設計解は, 図 6.11(b) に示す構造であった。

この実験では, すべての組合せが 76 通りと少なく, 構造が簡単であるために, 評価の幅が小さい。GA の大きな特徴として解候補空間すべてを探索する必要がないという点



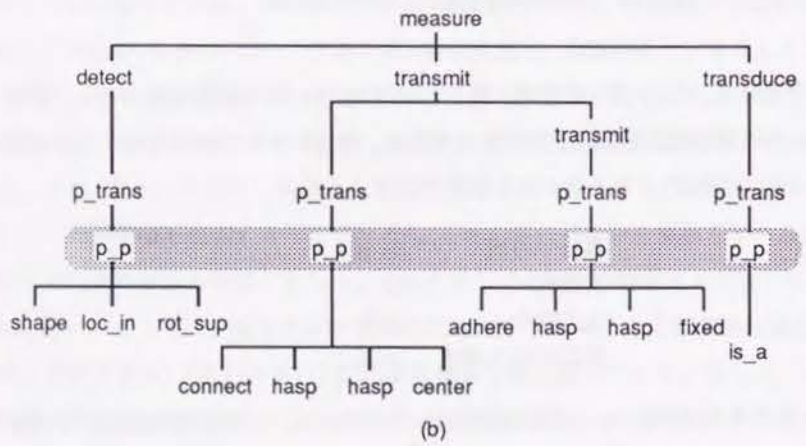


図 6.11: 出力結果とその概念図

が挙げられるが, 本実験においては, 解候補空間が狭く, GA の導入による効果は少なかった。より複雑な構造をもつ現実の設計に対しては, GA の導入が効果的であると考えられる。

今後の課題

ここまで述べてきたシステムは, 以下の点を考慮することにより, 概念設計解候補生成システムとしてより有効なものに発展させ得ると考えられる。

騙し問題の判定 騙し問題 (deceptive problem)[74] とは, 文字どおり GA を騙す問題をいう。

> **騙し問題** 初期分布π_0が与えられたとき, 特定の状態 i に対する吸収確率は交叉確率αの関数であるので, これを$\pi^i_\infty(\alpha)$ で表す。
> このとき, つぎの不等式を満たす $0 < \alpha < 1$ が存在する問題を, **騙し的である**という。
>
> $$\pi^{opt}_\infty(\alpha) < \pi^{opt}_\infty(0)$$

上記の定義を具体的にいえば「GA に交叉という操作を導入すると最適解へ収束する確率が低くなる問題」を騙し的であるということになる。つまり, 交叉をさせずに突然変異だけでも十分収束してしまう問題は騙し的である。

本研究では交叉を行なわない突然変異だけを行なうシステムを作成し, 収束しないことを確認した。これにより, 騙し問題ではないことを経験的に明らかにしたことになるが, さらに解析的に証明するためには, 上記騙し問題の定義中にある関数 π を定義し, より確かな証明を行なう必要がある。

エネルギ追跡法の改良 本研究では評価法の一つとしてエネルギ追跡法を提案したが, 実際に実装しているのは力, 熱などのエネルギに関する部分で, 位置的移動のエネルギや, 情報の追跡は考慮していない。

たとえば "公理論に基づく評価" 法の例としてとりあげたガソリンエンジンにおいて, 「力を電気に変換して再び力に変換するのは無駄である」という理由でマイナス点を付けているが, 実際のディーゼル車などは, このようなプランを用いている。これは, いったん電気信号に変換することで, 速度などの制御が容易になるためである。したがって, 制御情報などの情報の流れを追跡することによる評価の導入により, さらに有効な評価が可能であると考えられる。

機能等の知識の記述形式　現在用いている記述形式では, その明確な記述法が設定されていないため, 都合のよい記述形式になっている。より汎用性のあるシステムのためには, 形式的な知識の記述法が不可欠である。

より高度な評価基準の設定　上記の記述形式の改良とともに, より高度な評価法, たとえば物理構造物における挙動推論などの導入が効果的であると考えられる。

6.3　設計物の情報伝達モデルに基づく設計支援環境

6.3.1　設計目標から設計解への情報伝達モデル

機能系統図を完全なものとするためには, 図 6.12に示すように, 対応する構造系統図を構成し, 両者を統合化して組み込む必要がある。

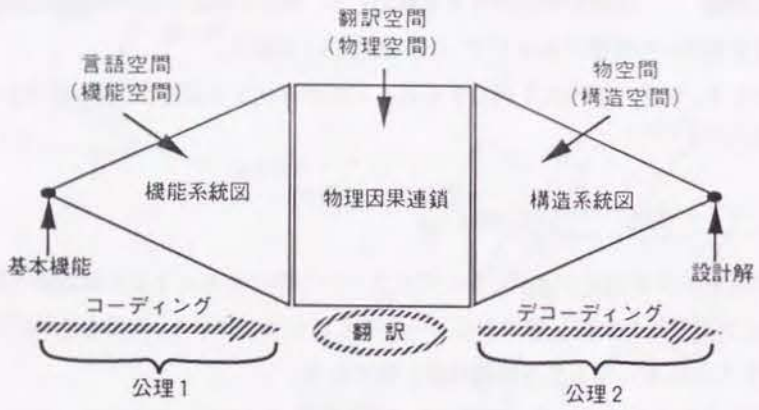


図 6.12: 情報伝達としての設計活動と設計公理

可撓棒の場合, この統合化された設計物表現は図 6.13のようになる。右半面に現れる構造系統図は, EBG システムで生成された説明木について, 部分要素を表す葉の部分を part_of 関係等に従って木状に構成したものである。図 6.12においては, 左端に設計者の目標を表すノード (基本機能; need) が, 右端には統合化された設計物全体を表すノード (設計物の統合的完成状態) があり, 中間にある**物理法則空間**, 左側にある**機能空間** (言語空間), 右側にある**構造空間** (物空間), がこれら二つのノードを継げる一種の媒介空間となっている。

このような設計物の深層構造は, つぎのように情報伝達のプロセスとして解釈することができる [17]。すなわち, 設計者の意図 (基本機能) が言語空間上で種々のコード (書き換え則, 対応則, すなわち, 機能・ブランに関連した一般・抽象知識) を介して順次コード化 (符号化) され, 最終的にはプリミティブな物理現象からなる因果連鎖を構成する。これが物理法則空間 (客観世界) 可読な形での設計者意図の “符号化 (表現)” を与えるも

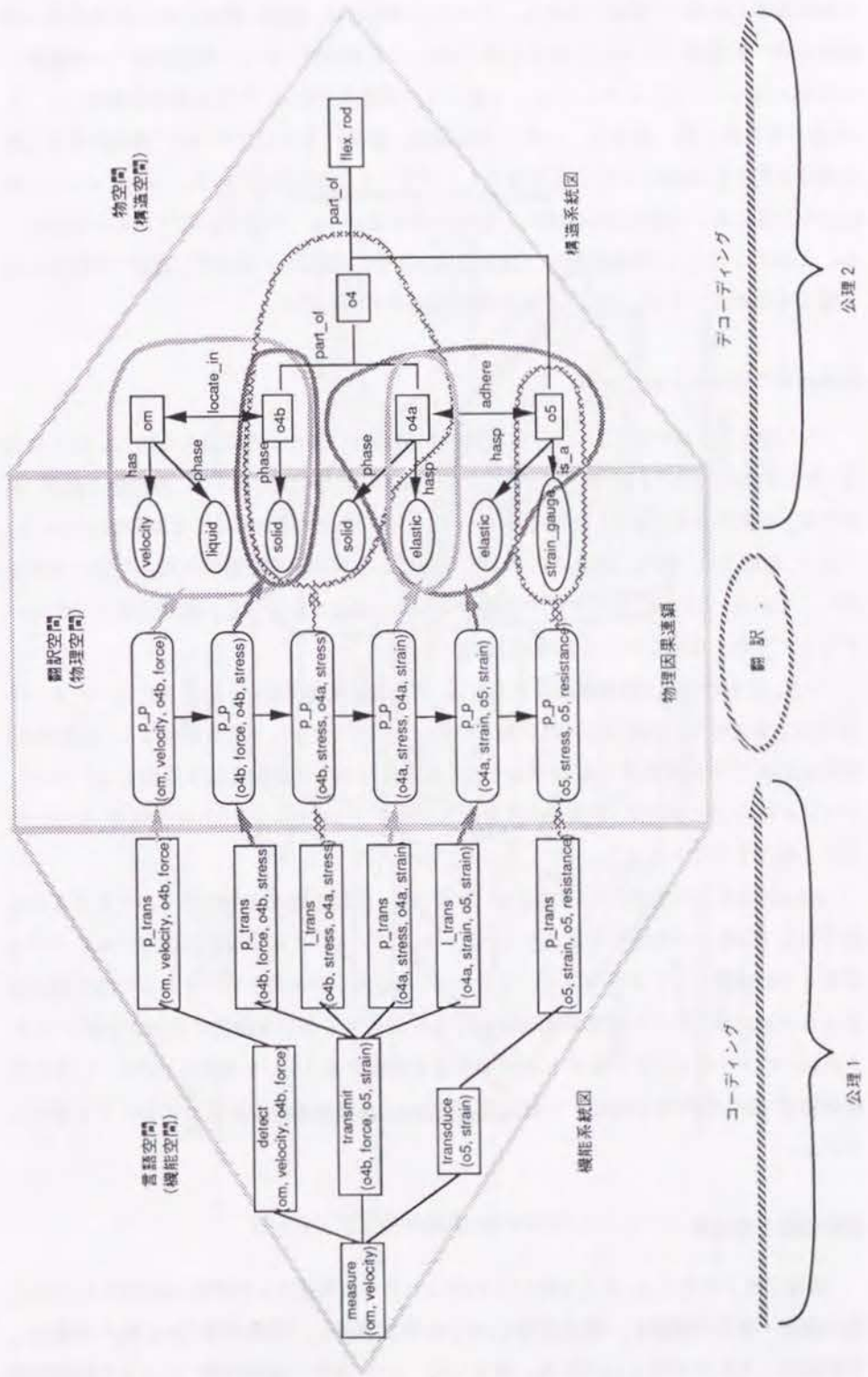


図 6.13: 可撓棒の機能–構造系統図

のであると考えられる。この物理空間上の因果連鎖は, その成立を支えるプリミィティブな物理構造の総体に “翻訳” される。これが基本的には, 意図→機能系列→因果連鎖→構造集合体への展開プロセスにおける最終端の, “言語空間” から “構造空間” への橋渡しの部分となっている。このようにして得られた構造集合体は, 今度は構造空間の “コード (構造の書き換え則, 統合則, 一般・抽象構造, 部品)” を用いて一つの “意味のある (基本機能を担う)” 組織化された構造体としてデコード (復号化) される。これによって, 図 6.12の左端にあった設計者の意図が, 右端の設計物として “伝達 (実現)” されたことになる。このように, 一つの設計物は “情報伝達 (符号化→翻訳→ 復号化) 過程” の結果として捉えることができる。以下, これを**情報伝達モデル**と呼ぶ。

情報伝達の合理性と設計公理

この情報伝達モデルにおいて, Suh の設計公理はつぎのように捉えられる。第 1 の公理 (機能実現の独立性) は, 図 6.12 の左部分にあたる言語空間において, 設計者の意図 (基本機能) が物理因果連鎖として正しくコーディング (符号化) されることを要請している。つまり, 機能間の干渉は, 機能それぞれの背後にある物理因果連鎖間の相互干渉の発生を意味しており, これは, それぞれの機能が正しく (成立するように) 因果連鎖としてコーディングされてはいないことを意味している。

一方, 第 2 の公理 (情報量の最小化) は, 物空間 (構造空間) におけるデコーディング (復号化) を含めて, 情報伝達過程が最もシンプルに (したがって最も効率よく, 経済的に) 実行されることを要請するものであり, これは復号化の元になる因果連鎖としてのコーディングならびに物空間 (構造空間) で使用可能なコード (部品など) の多様さ, 豊富さに大きく依存することになる。

2.3.2節の系 7 の説明において指摘したように, 公理 2 のほうが公理 1 よりもより根元的である (公理 2 が公理 1 を含意する) といわれている。これを情報伝達モデルの上でつぎのように解釈することができる。すなわち, 機能間干渉の発生からそれらの独立性を回復するためには, 互いの物理因果連鎖の干渉を防止する余分な構造の付加が必要となり, これは, デコーディングの際にも余分の要素を考慮すること, すなわち, デコードされた最終結果 (設計物の構造構成) の情報量 (複雑さ) を増加させる要因となることを意味している。

設計知識と情報量

情報伝達プロセスを, そこで用いられたコードとの関連でより詳細に把握するために, 先の機能・構造系統図を, “統合化軸 (part-of 軸)” 以外に, “抽象化軸 (is-a 軸)” を導入して詳細化したものが図 6.14である。図 6.15に, この “統合－抽象階層” による可撓棒の表

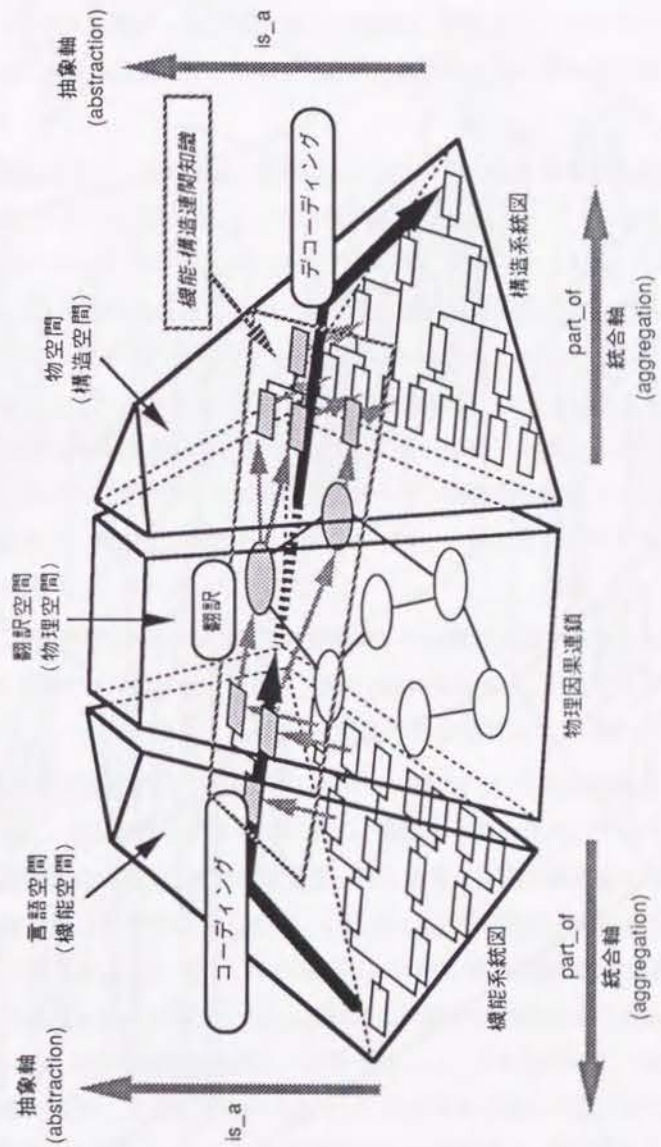


図 6.14: 設計物の抽象–統合モデル

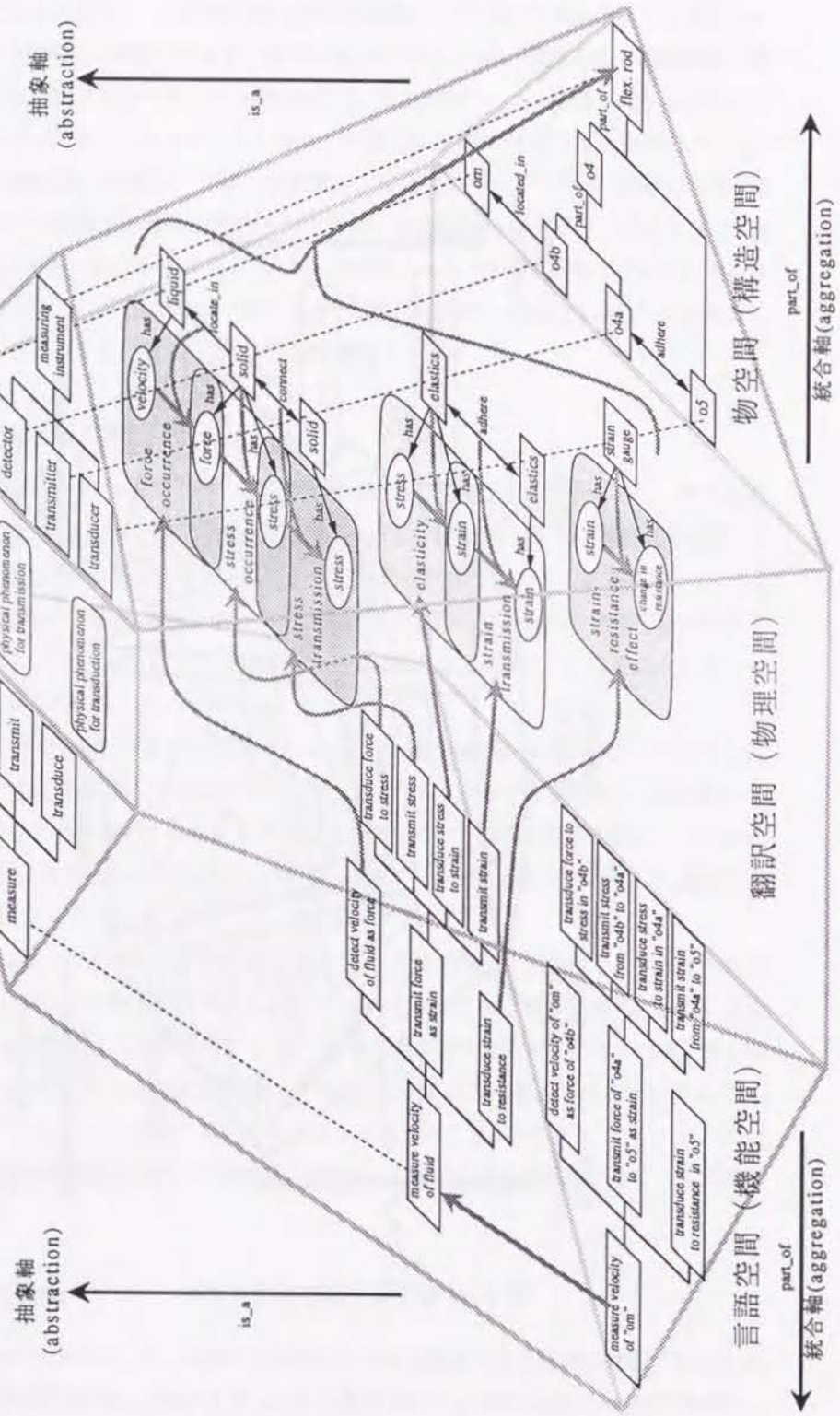


図 6.15: 可撓棒の抽象–統合モデル

現を示す。

説明木においては, 統合 (aggregation) と抽象 (abstraction) が混在して表されていた。たとえば,「X の材質は石英ガラスであり, 石英ガラスは弾性性質をもつから, X は弾性性質をもつ」という抽象軸に沿った推論と,「X は Y の部分であり, Y は Z の部分であるから, X は Z の部分である」という統合軸に沿った推論は, 説明木としては同一の構造であった。図 6.15ではこれらを区別して, 可撓棒の機能－構造系統図を立体的に表現したものである。

左半分 (言語空間) における設計意図 (基本機能) のコーディング過程の最適化を考えると, 最終的符号列, すなわちプリミティブな物理因果連鎖へのコーディングの際に, できるだけ抽象度の高いコード, たとえば抽象的プラン知識コード (measure → detect + transmit + transduce) などを多段階に組合せながら段階的に詳細化・具体化してゆく必要がある。すなわちコーディングのプロセスは抽象軸の最下層となる具体的詳細レベルの空間上で終始行われるというよりも, 図 6.14の太い矢印に沿って一旦上方に向い, 抽象レベルのコードを活用しながら段階的に統合軸の最下層の詳細・具体レベルへ移行してゆくと考えられる。同様に, 右半分 (物空間) においても, 抽象的な構造コードを活用しながらデコード (復号化, 統合化) が行われ, したがって, 一旦上昇したのちに下降する形で右端最下層の設計物ノードへ向うことになる。

設計者の意図 (基本機能) から設計解 (統合設計物) への情報伝達プロセスに対する情報論的分析を, より明確にするため, 図 6.16に, 課題 (基本機能達成) に対する可能解の領域が, 情報の付加 (“コードの適用”) によって, どのように限定されていくかを示す。

図の上半部, すなわち機能空間 (言語空間) においては, プラン, 抽象知識などの抽象コードの適用によって, 順次解領域が絞られてゆく。最終的に導出されたプリミティブな物理因果連鎖は, 物理法則空間 (因果合理性空間) 内にある “物理・構造複合体 (物理法則とその成立を背後で支える抽象構造の総体)” を媒介として, 構造空間 (物空間) における新たな解領域へと置き換えられる。これが物空間における抽象知識, 構成知識, 実装化知識などのコードの適用によって順次絞られ, 最終的には設計解の領域を与えることになる。この領域の狭さが, その情報量に対応し, 公理 2 からは, できるだけ広い領域を残すようにしなければならない。より厳密にいうならば, 公理 2 の立場からは, この領域の狭さは, 領域を絞るために適用されるコードの選択のための情報量と, 適用に際してコードを例示 (instantiate)・具体化する際に必要となる情報量に連関する。なお, 最下段の領域をできるだけ広くとるためには, 大づかみな論法が許されるならば, 前半の言語空間の段階から, できるだけ情報量 (制約量) の付加を避ける形でコード適用 (推論) を実行していく必要がある。これは先にも述べたように, できるだけ抽象度の高いコードの適用を示唆している。

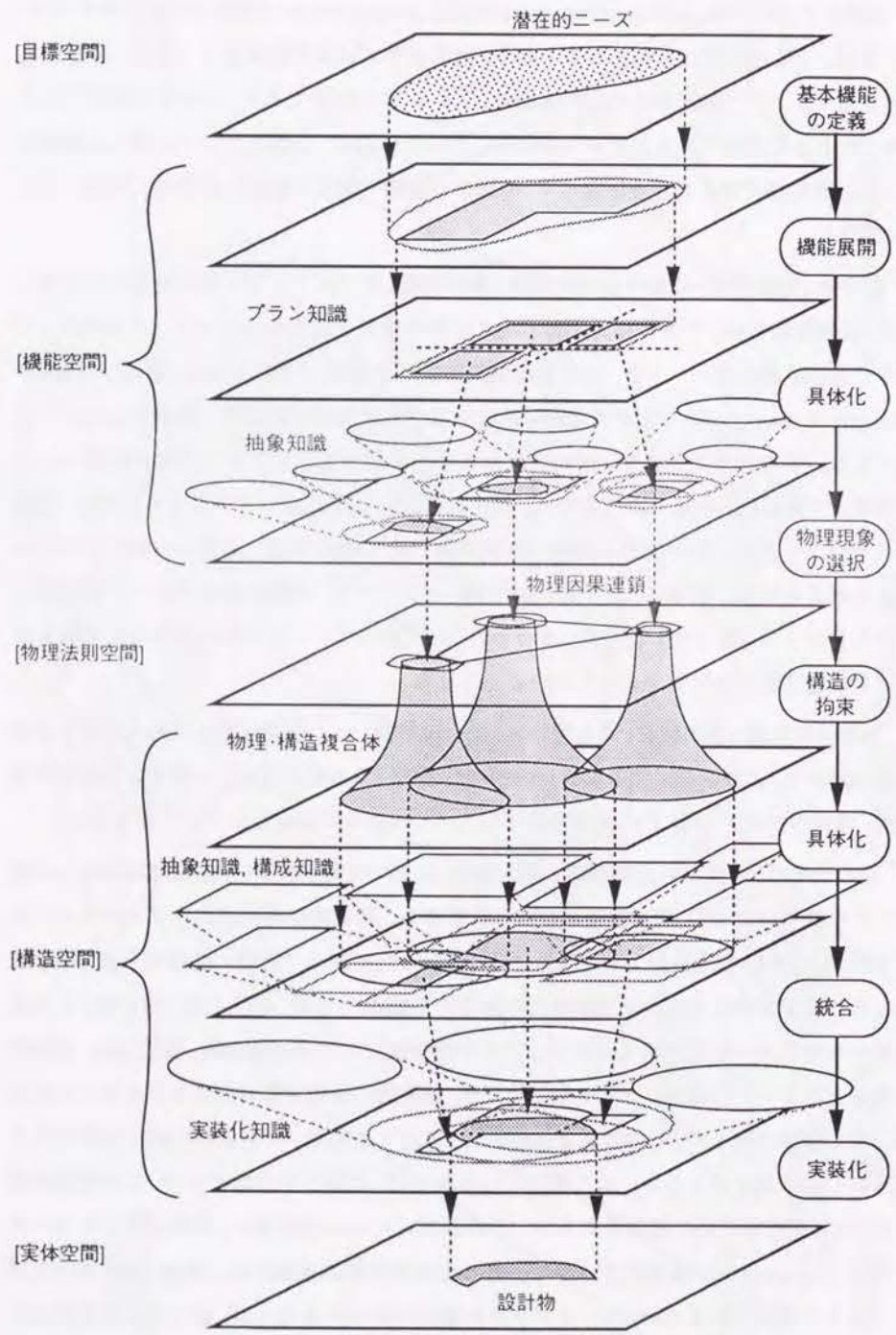


図 6.16: 情報付加過程としての設計プロセス

情報伝達モデルにおける獲得知識

上に述べたように, 設計公理論の見地から良好な設計を行うには, できるだけ抽象度の高い設計知識, すなわち“コード”を保有している必要がある。

4 章で導入した, EBG 手法に基づくチャンク化 (コンパイル) された知識の抽出は, ここで考えたコード化, デコード化による情報伝達の視点からは, マクロコード (統合コード, コンパイルされたコード) の抽出・獲得として位置づけることができる。このマクロコードの導入によって, コード化, デコード化の手順が簡略・短縮化され, 情報伝達の効率, 換言すれば設計作業の効率の向上が期待される。ただ, いたずらにマクロコードを導入することは, 参照すべきコードを増やすことになり, コード選定のための余計な情報量 (制約情報) の導入を必要とすることになる。

勿論, ここに述べた設計物の通信路・情報伝達モデルとしての解釈は, 設計過程を無駄や後戻りのない理想的なものに置き換えて捉えたものである。つまり, 現実の設計活動の際に生起しているコーディング, デコーディングは, このような一方向的な (コーディングの完成後にデコーディングが行われる) ものではなく, 両者が行きつ戻りつ実行されることによって設計物が具体化・詳細化されるとともに, 推論の失敗による後戻り, いわゆる試行錯誤, が頻繁に行われるものとなる。

このような言語空間, 物空間のインタラクション, すなわち相互干渉的推論として興味深いものに Clancey によって導入された Heuristic Classification [79] がある。これは基本的には診断などの分類型の問題を解決する際に, 人間 (専門家) あるいはエキスパートシステムが採用していると見られる推論戦略であり, 二つの概念階層, たとえば診断型 (原因追求型) 問題解決の場合, 原因事象の概念階層と観測される結果事象の概念階層の間で, 図 6.17 に示すように相互の関連性 (相関性) の最も高い, 適切な抽象度レベルで結果情報 (抽象化された結果事象) から原因事象候補 (抽象原因事象) へと, 連合的知識 (associational knowledge) を活用して移動し, そののち, 下方へ向かって詳細化・具体化が行われる。

これを設計の立場から見直すと, 結果事象空間は機能空間に, 原因事象空間は構造空間に対応し, これら両空間がそれぞれ抽象－具体レベルで階層化されており, 機能と構造の関係が最も密接に対応するレベルで, 機能空間上での抽象化推論 (抽象コードの適用) から, 構造空間 (のある抽象レベル) に移行し, 構造空間上での抽象・一般コードの組合せ適用による詳細化・具体化によって, 設計解 (設計物の構造構成) が求まることになる。

このような機能－構造の対応関係を規定する一つの大きな要因は, 媒介となる翻訳空間である物理法則空間の構造・特質である。しかし, この空間は, 種々の物理法則の規定・表現を見れば分かるように, 最もプリミティブな構造・機能によって最も抽象的に規定されており, この対応関係の適切なレベルを見い出すことは一般に困難となる。

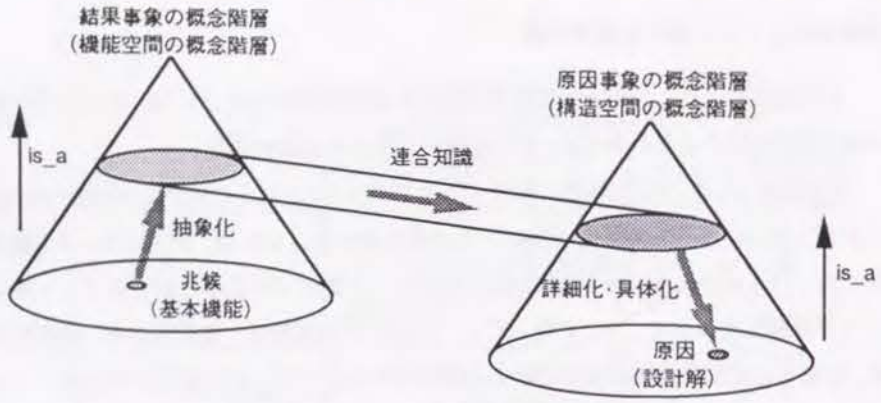


図 6.17: Heuristic Classification

他の要因としては, より上位にあるシステムとしての物理現象・構造, すなわち, 設計物の世界・空間のもつ特性, 換言すれば媒介となる物理法則空間だけでなく, その両側にある機能空間, 構造空間, ならびに設計物のシステムとしての特質が考えられる。これらは, 設計公理, すなわち公理 1 (機能の独立性) , 公理 2 (情報量の最小化) によって規定されており, この二つの公理をベースにして, 機能－構造の連合知識を抽出・獲得していくことが考えられる。5 章で提案した深い説明構造の導出は, そのシステマティックな方法の一つとなる。たとえば, 可撓棒の例で得られた深い説明構造 (図 5.2) は, 3 つの機能「流速測定」, 「耐熱性」, 「測定の局所性」と 3 つの構造要素「歪ゲージ」, 「棒状要素…長さをもつ要素」, 「棒状要素を包む筒管」の間の必然的, 緊密な関連性を導くものであり, モジュール化された機能 (群) －属性 (群) 連関知識, すなわち情報伝達過程におけるユニットコードを形成しているものとして解釈することができる。

6.3.2 ハイパーテキストを用いた設計支援環境

一般に, 設計支援に利用可能な情報は, 様々な働きと, 様々な形態をもち, お互いに複雑に関連しあっている。これらの設計知識を, 設計作業あるいは設計物の使用の際に活用することを考えると, 動的に移り変わる設計フェイズや設計者の意図に合わせて, 膨大で複雑に関連し合う知識空間の中から, 適切な設計知識を提示する設計支援環境が必要となる。そこで, 本章では設計支援のためのメディア環境としてハイパーテキスト (HYPERText) システムのあり方について検討する [6][16][80]。

ハイパーテキストによる設計情報参照

ハイパーテキスト環境は, ノードと呼ばれるまとまったテキスト (あるいは他の情報) の断片と, ノード間を結合させるためのリンクから構成され, 全てのノードはリンクをもつ [81]。このように, 整理・統一化された形式で様々な種類の情報を表現し, また, ノードに格納される情報に対して非線形かつ多様な参照形態を可能にするハイパーテキストシステムは, 設計支援の環境として良好な特性をもつといえる。たとえば, 異なった分野 (製造, 設計, 販売など) に携わるメンバが共同で設計を行なう, グループウェアのツールとしての有効性も指摘されている [82]。また, この環境は, 設計者から仕様者への正確で効果的な設計意図伝達のための良好なインタフェースとしても利用可能である。

一般に, ノードに格納される設計コンセプトの表現形態としては, テキスト, ネットワーク, ダイヤグラム, リージョン, テーブル, アナログ (図, 写真) などが考えられる。ノードを連結させるリンクのもつ参照形式としては, 構成–要素間関係, 抽象–具体関係, 集団–部分 (要素) 関係, クラス–インスタンス関係, 全体–部分関係, 原因–結果関係, 統合–分解関係, 実現–背景法則 (原理) 関係, 目標–手段関係, 複数の要因間の交叉関係などが考えられる。ノードの表現形態とリンクの参照形式とは密接な対応関係をもっている。たとえば, 部分–全体関係はリージョン, 要因間の交叉関係はテーブル, 構成・要素間関係はネットワーク, 実現はアナログ, 背景法則 (原理) はテキストで通常表示される。

情報伝達モデル上での設計概念形成

ノードに格納された情報が秩序を成し, 全体として一つの設計概念を形成する形での設計支援を考える。そのために, 情報伝達過程としてモデル化した設計プロセスを秩序形成の場とする。視覚的には, 図 6.14に示した抽象–統合の 2 つの軸をもつ空間内に配置された台形状の情報伝達経路 (以後 台形モデルと呼ぶ) 上に, ネットワークまたはテキストの形態をした設計情報 (以後 カードと呼ぶ) を分散的に張り合わせ, カードが連関構造を形成し, 全体としての設計概念となる (図 6.18)。カードの連関には, 物理因果関係を格納したカードが, カード要素変数の単一化を通して連鎖をなす場合も含まれる。このとき, 整理され秩序だったシンプルなカード集合体として表現される事が, 設計作業の容易さと関係する。チャンキングによるマクロコードは, その意味で重要である。

設計する立場 (情報発信者) から, 設計物を使用する立場 (受信者) へ視点を転ずるときにも, このハイパーカードシステムは有効な情報環境を提供するものと考えられる。すなわち, 台形モデルは, 発信者, 受信者何れの側にとっても情報伝達として普遍的なモデルであり, 個々のカードに内包される情報も普遍的なコンセプトを表すものである。ただ, これらカードの読み取り方, あるいは参照の仕方には, 当然のことながら変化が生じる。すなわち, カード情報は, 宣言的よりも手続き的に解釈する必要があろうし, また, 設計物の使用の根拠付け, 因果関係, 目標–手段関係, 全体–部分関係などに主として参照が行なわれることになる。勿論, 設計者と使用者では, 背景知識のレベルが全く異なるとも考えられるが, ここで導入した台形モデルの普遍性をより積極的に活用する立場からは, 設計支援に際して, 多数の背景知識の異なる専門家の協調設計作業を支援するグループウェア

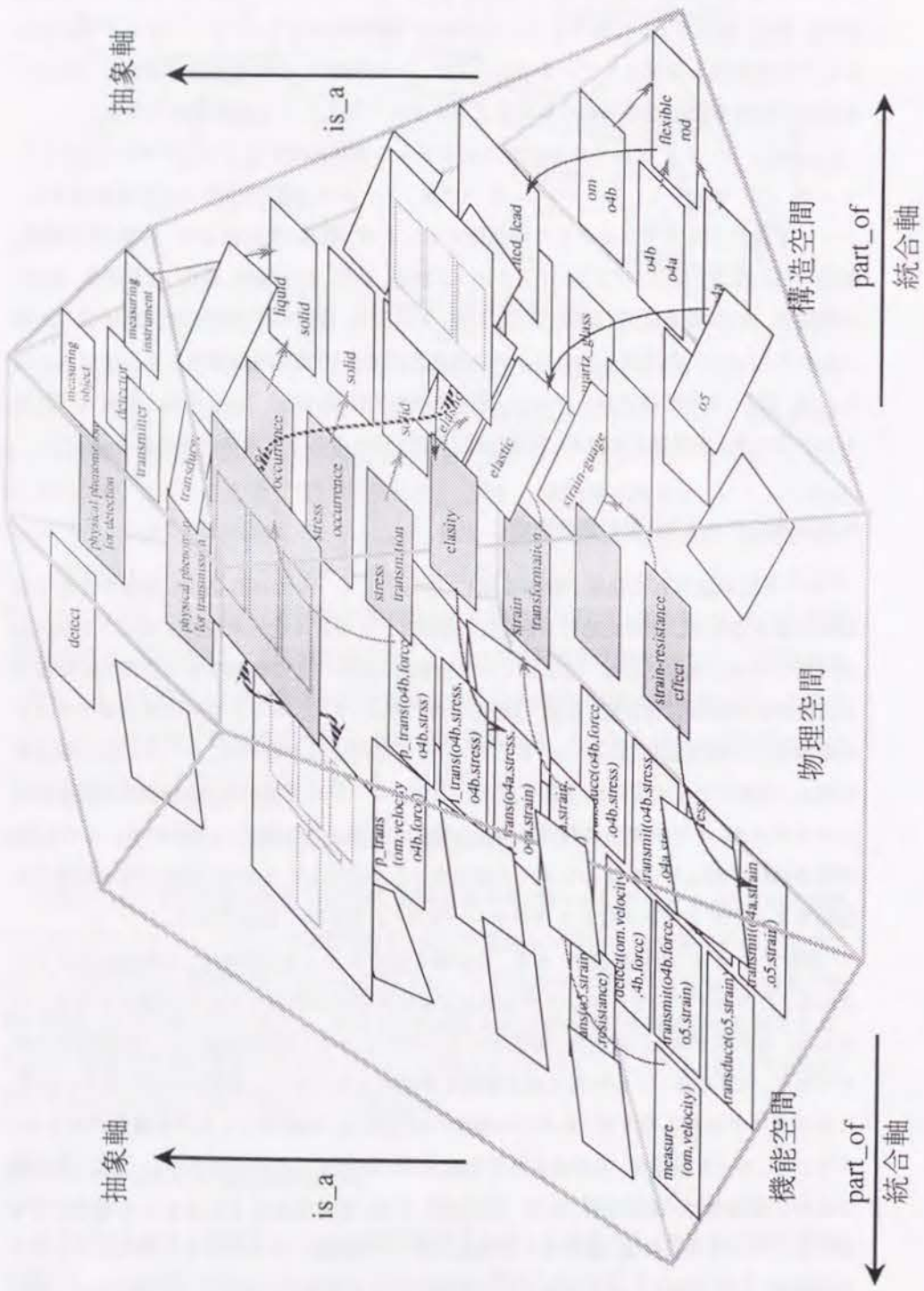


図 6.18: カードシステムの秩序形成

としてカードシステムを位置付けることになり, 必ずしも背景知識の斉一性を前提とすべきでないと考えられる。

本研究では, 最もポピュラーな Mac 上で稼働するハイパーカードを用いてシステムの構築を行なっている。まず, 比較的粒度の小さいコンセプト群をカードとして格納するとともに, 実体図などはイメージスキャナから取り込みカード化した。カード間の参照関係はボタン情報として, ハイパートークを用いてスクリプト表現しシステム内に組み込まれている (図 6.19)。

ハイパーテキストでよく問題にされる視点の喪失に対しては, 現在どの辺りに設計者の注視点があるかを表示するブラウザとして台形モデルを位置付けることを検討している。そのためには, システム側で設計者が参照あるいは構築 (例示化) したカード群が台形モデルの中で相互にどのように関連付けられて配置されるかを把握することができる知的情報処理能力が必要となる。これについては, EBG システムを稼働させている Prolog システムの活用を考えている。

設計知識選択の基準

一般化機能系統図をチャンク化して得られるマクロコードは, 台形モデル上で基本機能から設計解までをつなぐカードとして用いられる。このチャンク化としては, 4.6節で提案した設計知識評価尺度から考えると, 多くの組合せを構成する (適用範囲が広い) ことが期待される (新規設計を期待することができる) ような短いカードと, 逆に組合せの可能性は低いが伝達プロセスを大きく進めることが期待される長いカードが考えられる。また, 設計者の新規アイデアが盛り込みやすい汎用のカードと, 堅実に伝達プロセスを進める専用のカードという分類も考えられる。

このような様々なカードの中から, 設計者が目的に応じて適切にカードを選択することができるためには, カード群がその長–短, 汎用–専用の関係でわかり易く整理されている必要があり, 4.6節で示した図 4.16の導入が有効と考えられる。

6.3.3 設計情報空間

上述の秩序形成は, 設計目標に則してコンセプト群を統合表現するものであるが, その際に付随して参照されるべき, 設計事例の実体図, 予想される随伴事象, 背景原理 (法則), 要素間の干渉, あるいは可能代替案などの関連情報もハイパーテキストシステムに格納しておく必要がある。この干渉あるいは代替案のように一般に交叉関係として表示すべきものは, 先のチャンキングのような単一事例から抽出されるコンセプトではなく, 多数の設計物解析におけるチャンキングを集積することによって, 初めて導かれるものである。また, 図 5.2 や 図 5.8 のような, 設計コンセプトの背後にある五つの合理性 : 目標,

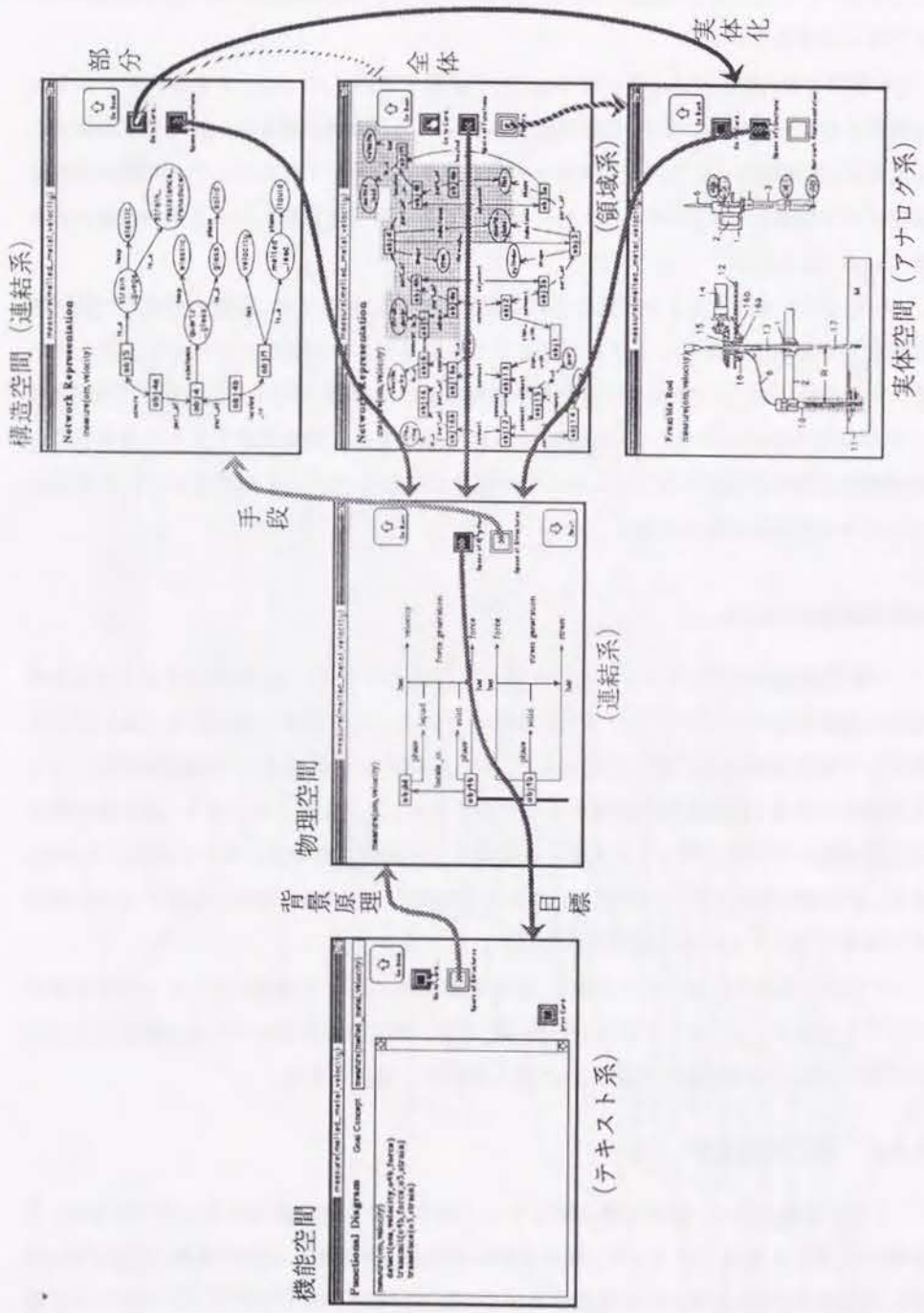


図 6.19: ハイパーカードシステムの構成

因果, 経済, 情報伝達 (公理 1, 公理 2) の見地からのマクロコンセプトをどのようにノードに格納し参照するかが, 今後の重要な課題であると考えている。

本研究では, 秩序形成の際に参照される設計情報として, 前章までに以下の 3 つの設計対象表現を提案している。

CD 理論による機能表現 概念依存理論 (Conceptual Dependency Theory) による機能表現動詞の解析により, 各々の動詞の概念を図式化することができる。この図式は, 機能のもつ意味の違いや意味的包含関係を視覚的に判断することを容易にする。これにより, 膨大な自然言語動詞の中から機能表現動詞の候補を選びだし, 意味的包含関係によって整理, 規格化することが可能になる。

意味ネットワークによる構造表現 設計論における認識公理や定性物理における NFIS 原理を考慮に入れ, 物理的設計物の構造を, 属性値の組で表された構造素と, それを結ぶ構造間関係で表現する。この場合, 設計物の構造は, 構造素のラベルと属性値をノードとし, 属性ラベルと構造間関係をアークとしたネットワークで視覚的に表現することができる。

物理因果連鎖による物理的設計物の把握 物理的設計物の機能の成り立ちは, 物理因果連鎖で説明することができる。物理因果関係には二つの一般型を考えることができる。個々の物理因果関係を一つのカードとみなすと, 二種類に統一された規格のカードの間で空きスロット (物理因果関係の変数) に同一の構造素を当てはめることによって, 因果連鎖が生じる。

4 章, 5 章では, 以上の表現形式を利用し, つぎに示す二つの設計知識獲得法を提案した。

EBG による一般化機能系統図自動生成と知識獲得 EBG による機能と構造の関係に着目した設計物の説明生成過程は, ハイパーテキストシステムのターミノロジーを用いると, つぎのように解釈することができる。すなわち, この過程は, 機能を副機能に展開する知識を用い, 基本機能を満たすように物理因果関係を表すカードを並べてゆくとき, 解析対象事例がそれによって生成される因果連鎖実現のための構造的条件を満たしているかのチェックの繰り返しとして実現される。

この説明の履歴である一般化機能系統図をチャンク化することによって設計複合知識 (マクロコード) が求められる。これらマクロコード群が豊富に存在する方が, 情報伝達過程としての設計作業を容易に進めることができる。しかし, 逆に大量のマクロコード群を維持・管理・運用, とくに検索する際には膨大なコストが予想されるため, 実際には参照頻度などを考慮した情報量評価により, マクロコードとしてチャンキングする価値がある

か否かを判断する必要があると考えられる。すなわち, このようなマクロコードを, 推論によりその場で直接導くコストと, マクロコードとして獲得しておき必要に応じて検索するコストとの比較検討が必要となる。この視点から, ハイパーテキストシステムによる計算機支援環境は, マクロコードを予め獲得しておく場合のコストを下げる試みとして位置付けることができる。

公理論的設計法に基づいた設計意図解析　解析対象事例に暗に埋め込まれた設計者の意図を表層化する“深い説明”は, EBG の出力と, ネットワーク表現された解析対象事例, 設計公理論に基いた設計属性の追加などを利用して生成される。その結果は, 複数の機能間の干渉とその解消方法を示すものである。これは, マクロコードの使用を抽象度の高いメタプランレベルでガイドする知識であると解釈することができる。

以上のように, 各々の手法はお互いに緊密な関係をもち, そこで使われる設計情報も, 図 6.20に示すように, 互いに複雑な参照が行なわれ, それゆえ, これらの設計支援環境の利用を繁雑なものにしている。そこで, 個々の設計情報は図式的に表現されるものが多いことに注目し, これらをハイパーテキスト環境に移植すれば,

- 設計情報空間のナビゲーション
- 時機を得た有用な情報の提示
- 関連する情報の活性化

などが可能になると考えられる。図 6.20中にカードとして示すのがハイパーテキストのノード, それらを結ぶ曲線がリンクに対応する。

6.3.4　今後の展望

上述の設計知識空間内に配置されたカードは, 設計者によって設計案形成時に, 台形モデル上でリンクをたどりながら任意に参照される。今後の展望としては, 各局面でどのような参照の形態があるのかを明らかにし, より有効な活用を図る必要があるものと考えられる。その一つの試みとして, 論証分析の考え方を参考にした設計情報の参照形態について検討する。

論証分析とは, 推論の過程や論点を文章単位で吟味することによって, その悪構造問題を対象とした議論に論理的な思考の枠組を導入するものである。論理分析とハイパーテキストは個別に考案され, 発達したものであるが, 1980 年代から両者の関連が注目されている [83]。論証分析における基本的な思考単位として, 主張 (claim), 根拠 (data), 理由づけ (warrant), 反論 (rebuttal) が知られている [83]。これらを, プラン (基本機能 (GC)

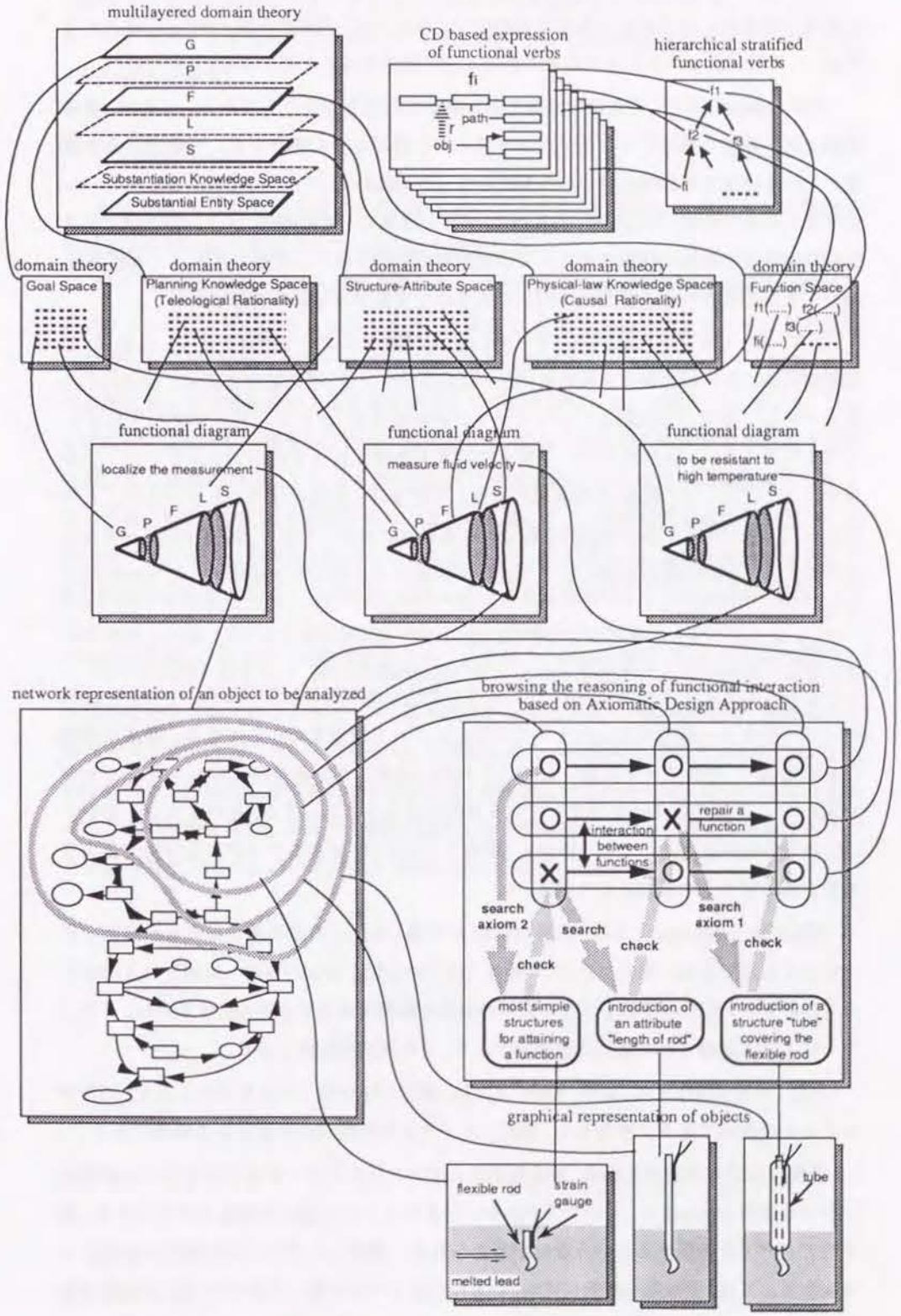


図 6.20: 設計支援に利用可能な情報とその間の参照関係

を物理因果連鎖にコーディングするためのマクロコード) の選択問題 (過程) に導入する場合, ハイパーテキストシステムをつぎのように活用することができる。

主張 (claim) とは, 何らかの目標を示し議論の口火を切ることである。これは, 基本機能 (GC) 達成の基本プランをプランリストから選ぶことに相当する。プランは基本機能とそれを達成する副機能列という形式をもった知識の塊 (チャンク) である。ハイパーテキストのもつ性質 (情報隠蔽) によって, プラン実現のための構造, プランの背景にある物理法則などの詳細な情報の混入による注視点の拡散を防ぐことができる。この性質は, 構造にとらわれない抽象化したレベルでの思考を支援する場合に有効である。

プランは, 4 章で示した手法のモード I (図 4.7参照) によって獲得される。これは, 初期領域知識にあらかじめ与えられていた一般的なプランを, 事例記述によって特殊化した記述である。多数の事例解析によってプランが集積され, プランリストが拡充されると, プランリストを, 一般的なプランを最上位に置き事例に特化されたプランをその下に配置することによって, 汎用–専門関係に注目して階層状に組織化することができる。この階層は, 4.6節で示した設計知識評価軸における横軸に対応しており, 新規な設計か堅実な設計か, という設計者の意図に応じて適切な階層レベルの知識を提示することが可能になる。

根拠 (data) とは, 主張 (claim) の拠り所となる事実を指す。すなわち, 選択したプランが良いことを支える事実は何か, どのような情報を拠り所としているか, という情報を, ハイパーテキスト環境のリンクをたどることによって検索することになる。確実な根拠 (data) として, そのプランを用いて GC を同時に達成している事例, または, そのプランを用いて GC の必要条件を達成している事例が考えられる。また, そのプランを用いて GC の十分条件を達成している事例も, 確実ではないが, 新たなアイデアを付加できる可能性があるプランの根拠となり得る。

理由づけ (warrant) とは, 根拠 (data) が主張 (claim) の拠り所であるとの解釈を正当化する記述である。理由づけは, 一般に「自然の法則」などの原理・原則という形をとるものが多い。過去の事例をプラン選択の根拠と解釈することを正当化するのは, そのプランによって展開された物理因果連鎖であり, また機能系統図である。

反論 (rebuttal) とは, 主張, 根拠, または, 理由づけが当てはまらないことや, 例外があることの表明である。すなわち, 選択したプランの再検討が必要となる場合である。

新規な設計を意図する場合, 階層化されたプランリストの中からできるだけ一般的なプランを選択 (claim) し, アイデアが出ないときはリンクを辿って階層を下げてゆき, 過去のアイデアを参照するという戦略が考えられる。選択したプランの再検討が必要となる場合としては, 予め基本機能 (GC's) に入れておくべきであった新たな GC が発見されたとき, claim の段階で別の選択枝を選ぶべきであった根拠が示されたとき, または claim

の段階を振替って新たなアイデアが出たときなどを想定することができる。この場合, ハイパーテキストシステムには, 台形モデル上を後戻りして別の選択枝を表示する機能を組み込む必要がある。

6.4 結論

本章では, 4 章および 5 章で提案した手法によって実在設計物から抽出された知識を利用して設計活動を支援する二つの方法を提案した。

第一の利用方法は, 特定の判断基準のもとで要求機能を満たす設計解候補を遺伝的アルゴリズムによって生成する方法である。本研究では, 物理因果連鎖上でエネルギの流れに注目した評価, 設計公理論に基づいた物理因果連鎖そのものの単純性による評価, および必要となる構造属性数による評価を導入して検討を行なった。当然のことながら, この 3 つは設計物全体を評価する基準の一部に過ぎず, 提案された設計解候補の総合的な判断は設計者に委ねられることになるが, "特定の" 判断基準下での設計解候補提案システムとしては, 本研究の提案は有効であると考えられる。また, 評価関数定義の柔軟性という遺伝的アルゴリズムの特性を活用することによって, 機能ならびに物理因果連鎖に対する直接的評価が可能になる。このため, 従来の組合せ最適化問題として定式化した設計プロセスモデルと比べて, より概念設計レベルに近い段階での設計解生成が可能となる。

第二の利用方法として, 設計の情報伝達モデル上で 4 章および 5 章で獲得した知識を分散的に張り合わせることによって, 基本機能から実体形成への経路を経て, 設計概念を形成する手法を提案した。この手法は, 計算機による自動設計を指向するものではなく, 設計活動の主体はあくまで設計者自身にあり, 計算機は時期を得た設計情報の提示システムとしての役割をもつ。そこで, 複雑に関連し合う設計情報をわかりやすく表示するインタフェースとしてハイパーテキスト環境について考察した。

第 7 章

結言

本論文は物理的設計物の概念設計段階を知的な形で支援する計算機環境についての研究成果をまとめたものである。ここで提案した内容を要約すると以下のようになる。

3 章では, 設計活動および物理的設計物のモデルを提案した。まず, 価値工学の分野で用いられている目的論的な設計対象の機能分析の視点から, 設計活動を, 基本機能 (G)・機能 (F)・構造属性 (S)・実体 (R) を連続的に選択する行為としてモデル化した。さらに, これらの選択の背後にある合理性を反映した知識, すなわちプラン型知識 (P), 物理因果関係 (L), 実装化知識 (I) を導入し, G, (P), F, (L), S, (I), R からなる**階層型設計知識空間**と, これによって記述される**設計対象物モデル**を提案した。提案したモデルは, 従来の CAD システムに見られるような幾何形状などの構造属性に注目したモデルとは異なり, 構造を規定する機能や, 構造と機能を結び付ける原理原則的な知識 (物理因果関係) を直接的に表現することができる。したがって, CAD におけるモデルが詳細設計段階の支援に有効であるのに対し, 提案したモデルは, 設計要求を満足する設計案をつくり出す過程である概念設計段階, 企画段階の支援に有効である。

設計活動および設計対象物のモデルを計算機上に記述し操作するための設計知識は, 場当たり的に与えると汎用性を失うことになる。そこで, 統一的な形式をもち, かつ設計の深層的な部分に立ち入って設計物を記述するための設計知識として, G, P, F, L, S の内容について検討した。とくに, 物理因果関係 (L) については二つの一般形を提案し, 物理現象に支配される設計物を物理因果関係の連鎖 (**物理因果連鎖**) で記述できることを, 測定器の場合を例にとり上げ示した。

機能表現の方法に関しては, 機能表現動詞に対して CD 理論に準じた解析を行ない, その意味内容 (**概念依存表現**) を明らかにすることによって, これら動詞間に意味的重複や包含関係を見い出し, これらを階層構造状に整理する方法を, 位置的移動を表す動詞を例にとり提案した。

構造の表現法に関しては, 一般設計学の分野で提案されている**認識公理**に基づいて, ネットワーク表現法を提案した。ここで提案した表層的かつ客観的な構造記述法は, 4 章

以降で提案した知識獲得および利用システムに対する入力となるものであり，システム操作者の入力データ作成の労力を軽減する意味でも，また構造表現に機能的内容を含めると機能推論の本質的な意味を失うという指摘 (定性推論における NFIS 原理) に沿う意味でも，妥当であると考えられる。

4 章では，EBG の説明木生成および一般化の枠組を用いて，概念レベルでの設計支援のための知識獲得システムを提案した。これは，3 章で示した階層的設計知識を EBG 手法における領域知識として設計事例を解析し，複数の領域知識と設計事例のもつ情報とを加えて一つの新たな知識にコンパイルする手法である。EBG 手法には，「知識獲得のための知識獲得」というパラドクスや，「生成された知識が真の意味での新しい知識ではない」という問題が指摘されている。これに対しては，提案したシステムでは，事例解析情報が加えられて初めて因果合理性が得られるように，領域知識に一般性をもたせることによって解決している。

この枠組みでは，領域知識は SIG(Single Instance Generalization) を正当化するために使われており，過大な一般性を与えてこの能力が失われないように注意する必要がある。そこで，含意命題形式で与えられている領域知識に対し，より収集が容易で，より一般性をもった記述形式である決定則を導入し，論理的な正当性を失わずに，一般化された知識から設計知識を獲得する手法を提案した。

また，階層構造状に整理された設計知識を用いて説明を行なうことにより，説明木に複数の操作性境界が出現する。これを利用して，一つの事例から多面的視点により種々の操作性知識を抽出する方法 (モード) を設定し，この獲得モードと関連させて，獲得された知識の質 (操作性) に対する検討を加えた。

5 章では，公理論的設計アプローチに基づいて，設計事例に対するより深い理解を得る手法を提案した。この手法は，まず，複数の機能を同時に満たしている解析対象事例の構造記述から，単一の機能だけを達成するための構造的条件を 4 章のシステムを用いて抽出する。つぎに，抽出された構造記述に対し，設計公理 2 に従った簡略化操作と，設計公理 1 に従った機能間の干渉の除去操作という 2 つの操作を交互に施す。これにより，複数の機能間の干渉が，設計事例においてどのような構造によって回避されてるかという，メタプラン的な知識が獲得される。

6 章では，4 章および 5 章で提案した手法によって獲得された知識を利用して，設計活動を支援する方法を検討した。

第一の方法は，特定の判断基準のもとで要求機能を満たす設計解候補を遺伝的アルゴリズムによって生成する方法である。要求機能を満たす物理的設計物を設計する問題は，まさしく設計型の問題に分類され，種々の困難を伴っている。このため従来は，設計物の大まかな構造を予め与えた上で，設計パラメータを最適化するという形で計算機支援が行

なわれている。本研究で提案した手法は，逆に詳細な設計パラメータや幾何形状にはとらわれず，設計要求を満たすことのできる物理因果連鎖およびそれを達成するために必要な構造属性を提案するものであり，従来のシステムと相い補う形の利用が期待されるものである。本研究では，3 つの評価基準を導入して検討を加えた。当然ながら，これらは設計物を評価する基準の一部に過ぎず，生成された設計解候補の総合的な判断は設計者に委ねられることになるが，“特定の” 判断基準下での設計解候補提案システムの枠組として，本提案は有効であると考えられる。

第二の方法として，獲得された知識を設計者が有効に利用するためのハイパーテキスト環境について検討した。まず，3 章で提案した連続的選択行為としての設計過程モデルを，情報伝達経路としての視点から解釈しなおした設計の情報伝達モデル (台形モデル) を提案した。このモデルにおいては，4 章で提案した EBG システムによって獲得された知識は，台形モデル上で設計目標と設計解をつなぐ断片的な通信経路として解釈される。また，5 章で提案した深い理解により得られたメタプラン的知識は，機能－構造連関知識 (モジュール化されたユニットコード) として解釈される。台形モデルにおいて，公理論的設計アプローチで提案された公理 1 は，機能空間から構造空間への「意思」の疎通が誤りなく行なわれることを，公理 2 はそれが最小の情報量によって最も効率よくなされることを要請する。したがって，ユニットコードは，設計の効率を向上するという意味に加えて，公理論によって合理性・最適性が保証されているという意味でも，非常に価値の高い知識であると思われる。台形モデルをこれらの断片的な設計知識の秩序形成場として，その上で獲得された知識が基本機能から実体形成への経路を形づくることによって，設計案が構成される。この手法は，設計の自動化を指向するものではなく，人間の設計活動を支援する計算機システムのありかたの一つの提案である。

以上のように，本研究では，第 3 章で提案した設計過程および設計物のモデルに基づいて，階層型設計知識空間を設定し，第 4 章でその知識を機械的に獲得する方法，および第 5 章でその知識の組合せをガイドするメタプラン知識を設計物から獲得する方法を提案し，第 6 章で，獲得した知識を利用して概念設計を支援する方法を提案した。本研究で提案した一連の設計支援環境は，設計の大まかな案を生成する概念設計段階に注目して支援する一つの形態として，従来から提案されている詳細設計段階に対する支援システムと相い補う形で利用が期待されるものと考えられる。

謝辞

本研究をとりまとめるにあたり, 懇切なるご指導, 御配慮を賜わり, 多大な御尽力をいただいた京都大学工学部 岩井壮介教授に心から感謝いたします。

京都大学工学部在学中に知識工学に関する研究を行う契機を与えて頂くとともに, 本研究を遂行するにあたり終始懇切なご指導, 御鞭撻を賜わった京都大学工学部 片井修助教授に深甚な謝意を表します。

また, 平素から有益な御教示と御指導を頂いた岡山大学工学部小西忠孝教授, ならびに 馬場充助教授に深く感謝いたします。

最後に, 本研究を遂行するにあたり, 御支援と暖かい御激励を頂いた樋木哲夫助手をはじめとする京都大学工学部精密工学教室の皆様, ならびに岡山大学工学部情報工学科の皆様にお礼申し上げます。

参考文献

[1] 赤木新介; 設計工学 (上), コロナ社, 1991

[2] 玉井正寿; 価値分析, 現代経営工学全書, 森北出版, 1978

[3] 渡辺大助; VE 理論の研究 –設計学的アプローチ, 第 17 回 VE 全国大会 VE 研究論文集, pp.117-124, 1984

[4] 片井修, 川上浩司, 椹木哲夫, 岩井壮介, 小西忠孝; 説明に基づく学習手法を用いた事例からの操作的設計知識獲得, 計測自動制御学会 論文集, vol.26, No.8, pp.916-923, 1990

[5] R. C. Schank, C. K. Riesbeck; Inside Computer Understanding, Lawrence Erlbaum, 1981

[6] O. Katai, H. Kawakami, T. Sawaragi and S. Iwai; Computer Assisted Value Engineering: A Standardized and Generalized Way of Functional Analysis Based on Artificial Intelligence Techniques, Proc. of Pacific Conference on Manufacturing '92, pp.782-789, 1992

[7] 吉川弘之; 一般設計学序説, 精密機械, **45**, 8, pp.906-912, 1979

[8] H. Kawakami, T. Konishi; The Systematic Method for Constructing The IDEA BANK Based on The EBL, Memoirs of The Faculty of Engineering, Okayama University, Vol.26, No.1, pp.95-108, 1991

[9] T. M. Mitchell, R. M. Keller & S. T. Kedar-Cabelli; Explanation-based Generalization: An Unifying View, Machine Learning, **1**, pp.47-80, 1986

[10] O. Katai, H. Kawakami, T. Sawaragi and S. Iwai; An EBL System for Acquiring Conceptual Design Knowledge from Value Engineering Perspectives, Proc. of Pacific Rim International Conference on Artificial Intelligence '90, pp.589-594, 1990

[11] S. J. Russell; Analogy and Single Instance Generalization, Int. Workshop on Machine Learning, pp.390-397, 1987

[12] O. Katai, H. Kawakami, T. Sawaragi and S. Iwai; A Knowledge Acquisition System for Conceptual Design based on Functional and Rational Explanations of Designed Objects, Artificial Intelligence in Design '91, pp.281-300, 1991, Butterworth Heinemann

[13] N. P. Suh; The Principles of Design, Oxford University Press, 1990

[14] J. Doyle; A Truth Maintenance System, Artif. Intell., **12**, pp.231-272, 1979

[15] D. E. Goldberg; Genetic Algorithms in Search, optimization, and machine learning, Addison-Wesley, 1989

[16] 片井, 増市, 椹木, 岩井, 川上, 松原, 井田; 秩序形式による問題解決と意思決定支援, 日本機械学会 RC106 研究分科会報告書, 1993

[17] 片井修, 川上浩司, 椹木哲夫, 岩井壮介; 機能から構造への写像空間における設計物表現と設計知識抽出, 第 10 回設計シンポジウム 講演論文集, pp.31-40, 1992

[18] CIM を指向する CAD/CAM と生産管理実務総覧, 技術資料センター, 1991

[19] 鈴木宏正, 他; プロダクトモデルに基づく幾何学的拘束関係の記述と寸法処理への応用, 精密工学会誌, **52**, 4, pp.1037-1042, 1990

[20] F. Kimura, et.al. ; Integration of design and manufacturing activities based on object modeling, Advances in CAD/CAM, p.375, 1983

[21] J. R. Dixon (村上 存 訳); Research in Designing with Features, Computrol, No.25, pp.81-90, 1988

[22] 日本設計工学会; CAD 標準化と STEP, 日本設計工学会, 1991

[23] P. Freeman and A. Newell; A Model for Functional Reasoning in Design, Proc. of IJCAI-71, pp.621-640, 1971

[24] K. T. Ulrich, W. P. Seering; Function Sharing in Mechanical Design, Proc. of AAAI '88, pp.342-346, 1987

[25] 赤木新介; 設計論とコンピュータ利用技術 (1), 機械の研究, **38**, 1, pp.1-6, 1986

[26] 赤木新介; 設計論とコンピュータ利用技術 (2), 機械の研究, **38**, 2, pp.273-278, 1986

[27] 石田智利; 概念設計システム, 機械の研究, **37**, 1, pp.199-202, 1985

[28] D. C. Brown and B. Chandrasekaran; Knowledge and Control for a Mechanical Design Expert System, Computer, 1986

[29] A. A. Araya and S. Mittal; Compiling design Plans from Descriptions of Artifacts and Problem Solving Heuristics, IJCAI 87, pp.552-558

[30] B. Faltings; Qualitative Kinematics in Mechanisms, Proc. of IJCAI-87, pp.436-442, 1987

[31] M. Dyer, M. Flowers, and J. Hodges; Edison : An Engineering, design Invention System Operating Naively, Proc. 1st Conf. of Applications of Artificial Intelligence in Engineering, Problems, pp.327-341, 1986

[32] D. Navinchandra; Case Based Reasoning in CYCLOPS, a design Problem Solver, Proc. of DARPA Workshop on Case-Based Reasoning, pp.286-300, 1988

[33] K. T. Ulrich, W. P. Seering; Conceptual design as novel combination of existing device features, Proc. ASME Design Automation Conf., p.295-300, 1987

[34] 奥野 拓 他; GA による設計問題へのアプローチ, 日本機械学会 ロボティクス・メカトロニクス講演会 '92 講演論文集, pp.387-390, 1992

[35] N. P. Suh, A. C. Bell and D. C. Gossard; On an Axiomatic Approach to Manufacturing Systems, Trans. of ASME, J. Eng. Ind., **100**, pp.127-130, 1978

[36] C. W. Bytheway; The Creative Aspect of FAST Diagramming, An advanced course of instruction for the Value Engineer, Proc. of the SAVE conference,pp.450-454, 1971

[37] ICOT KAD-WG 活動報告書, 1989

[38] J. Barber, et.al.; ASKJEF: Integration of Case-Based and Multimedia Technologies for Interface Design Support, Artificial Intelligence in Design '92, pp.457-475, Kluwer Academic Publishers, 1992

[39] G. Fischer, K. Nakakoji; Empowering designers with integrated design environments, Artificial Intelligence in Design '91, pp.191-210, Butterworth Heinemann, 1991

[40] 日本 VE 協会 機能分析技法開発研究会; 研究資料 2: 機能分析技法, 日本 VE 協会, 1983

[41] 日本 VE 協会 機能定義技法開発研究会; 研究資料 1: 機能定義技法, 日本 VE 協会, 1982

[42] 日本 VE 協会 東京支部 FAST 技法研究会; VE 資料 24: FAST マニュアル, 日本 VE 協会, 1976

[43] 玉井正寿 編; VE と標準化, 日本規格協会, 1981

[44] 日本 VE 協会 東京支部 VE 用語研究会; VE 資料 49: 機能用語の選定とその分類体系, 日本 VE 協会, 1981

[45] 中沢　弘; 公理論的設計法, 精密機械, **50**, 2, pp.341-346, 1984

[46] 中沢　弘; 情報積算法 –成功のための方法論–, コロナ社, 1987

[47] C. Lewis; Why and How to Learn Why : Analysis-based Generalization of Procedure, Cognitive Science, **12**, pp.211-256, 1988

[48] G. Dejong; Some Thoughts on the Present and Future of Explanation-Based Learning, Proc. of the 8th European Conf. on Artificial Intelligence, pp.690-697, 1988

[49] R. M. Keller; Defining Operationality for Explanation-Based Learning, Proc. of AAAI 87, pp.482-487, 1987

[50] S. T. Kedar-Cabelli & L. T. McCarty; Explanation-Based Generalization as Resolution Theorem Proving, Proc. of 4th Int. Workshop on Machine Learning, pp.383-389, 1987

[51] H. Hirsh; Explanation-Based Generalization in a Logic-Programming Environment, Proc. of IJCAI-87, pp.221-227, 1987

[52] 平野広美; 積木問題を遺伝的アルゴリズムで解く, インターフェース 2 月号, No.177, CQ 出版社, 1992

[53] Themsky; 鳶もいつかは鷹を生むか？, bit, **23**, 5, 共立出版, 1991

[54] J. J. Grefensteffe et.al.; Genetic algorithms for the traveling salesman problem, Proc. of an International Conference on Genetic Algorithms and Their Applications, pp.160-168, 1985

[55] H. A. Simon; The Sciences of the Artificial, MIT Press, Cambridge, 1982, (稲葉, 吉原 訳; システムの科学, パーソナルメディア, 1987)

[56] 岩井壮介, 他; 知識システム工学, 計測自動制御学会, 1991

[57] J. de Kleer; An Assumption-based TMS, Artif. Intell., **28**, pp.127-162, 1986

[58] 岩崎幸雄; 高温流体の流速測定装置, 特許公報 昭 60-10580, 日本国特許庁, 1985

[59] 秋山兼夫; FAST の研究, 日本 VE 協会誌, **100**, pp.46-52, 1984

[60] 片井修, 川上浩司, 椹木哲夫, 岩井壮介; 概念設計のための機能表現と推論システム, 第 31 回システムと制御研究発表講演会 予稿集, pp.243-244, 1987

[61] T. R. Davies & S. J. Russell; A Logical Approach to Reasoning by Analogy, IJCAI-87, pp.264-270, 1987

[62] 片井, 椹木, 岩井, 井田, 川上; 対象領域の順序構造に基づいた学習とファジィ推論, 日本機械学会 RC94 研究分科会報告書, 1991

[63] 谷口 修; 計測通論, 良賢堂, 1988

[64] O. Katai, T. Sawaragi, S. Iwai and W. Sue; An Intelligent Interface System for Analysis and Design of Sensing Devices, Proc. of IECON '84, pp.503-508, 1984

[65] M. S. Braverman and S. J. Russell; Boundary of Operationality, Proc. of 5th Intl. Conf. on Machine Learning, pp.221-234, 1988

[66] 川上浩司, 小西忠孝, 池本文典; 決定則を用いた機械学習による概念設計に対する知識表現の獲得, 第 10 回設計シンポジウム 講演論文集, pp.41-50, 1992

[67] 川上浩司, 池本文典, 近藤文剛, 小西忠孝; 決定関係の特殊化による設計知識獲得, 第 17 回知能システムシンポジウム 資料, pp.63-68, 1993

[68] 日本規格協会 ; JIS ハンドブック機械要素, 1984

[69] S. Rajamoney & G. Dejong; The Classification, Detection and Handling of Imperfect Theory Problems, Proc. of IJCAI-87, pp.205-p207, 1987

[70] W. W. Cohen ; Abductive Explanation-Based Learning: A Solution to the Multiple Inconsistent Explanation Problem, Machine Learning, **8**, pp.167-219, 1992

[71] 山村, 小林; EBL の複数例題下への拡張, 人工知能学会誌, **4**, 4, pp.389-397, 1989

[72] 片井修, 川上浩司, 椹木哲夫, 岩井壮介; 公理論的アプローチによる設計知識と設計物モデル, 人工知能学会 第 19 回知識ベースシステム研究会, 1992

[73] 池本文典, 川上浩司, 小西忠孝; 概念設計解候補創出のモデル, 第 32 回 SICE 学術講演会 予稿集, pp.591-592, 1993

[74] 小林重信; 遺伝的アルゴリズムの現状と課題, 計測と制御, **32**, 1, pp.2-9, 1993

[75] 山村雅幸, 小野貴久, 小林重信; 形質の遺伝を重視した遺伝的アルゴリズムに基づく巡回セールスマン問題の解法, 人工知能学会誌, **7**, 6, pp.117-127, 1992

[76] 伊庭斉志; 情報の進化と創造的学習について, 人工知能学会研究会資料, SIG-F/H/K-9001-10, 1990

[77] 村上存, 中島尚正; 機械の設計診断と人工知能, 人工知能学会研究会資料 SIG-KBS-9002-9, 1990

[78] 梅田靖, 富山哲男 他; 対象モデルに基づく定性物理を用いた故障診断, 人工知能学会全国大会 (第 3 回) 論文集 [I], pp.263-266, 1989

[79] W. J. Clancey; Heuristic Classification, Artificial Intelligence, **27**, pp.289-350, 1985

[80] 片井修, 川上浩司, 椹木哲夫, 岩井壮介, 京本洋次郎, 中村隆義; 伝達路としての設計物表現と設計・使用情報環境, 第 17 回知能システムシンポジウム 資料, pp.75-80, 1993

[81] J. Nielsen (斎藤 孝 訳); HYPER Text & HYPER Media, HBJ 出版局, 1991

[82] 松下温 編; グループウェア入門, オーム社, 1991

[83] R. E. Horn (松原光治 監訳); ハイパーテキスト情報整理学, 日経 BP 社, 1991